KATHOLISCHE UNIVERSITÄT
EICHSTÄTT-INGOLSTADT

DISSERTATION

From Bag-of-Words Towards Natural Language: Adapting Topic Models to Avoid Stop Word Removal

Author:
Max SCHULZE DIECKHOFF

First Supervisor:
Prof. Dr. Joachim BÜSCHKEN
Second Supervisor:
Prof. Dr. Thomas HOFFMANN

WFI Ingolstadt School of Management
Chair of BA, Distribution Management and Marketing

Date of Disputation: November 4, 2021

"*If nobody makes you do it, it counts as fun.*"

– Hobbes[1]

[1] of Calvin and Hobbes

KATHOLISCHE UNIVERSITÄT EICHSTÄTT-INGOLSTADT

Abstract

Chair of BA, Distribution Management and Marketing
WFI Ingolstadt School of Management

Doctor of Philosophy

From Bag-of-Words Towards Natural Language: Adapting Topic Models to Avoid Stop Word Removal

by Max Schulze Dieckhoff

Topic models such as latent Dirichlet allocation (LDA) aim to identify latent topics within text corpora. However, although LDA-type models fall into the category of Natural Language Processing, the actual model input is heavily modified from the original *natural language*. Among other things, this is typically done by removing specific terms, which arguably might also remove information. In this paper, an extension to LDA is proposed called *uLDA*, which seeks to incorporate some of these formerly eliminated terms – namely stop words – to match "natural" topics more closely. After developing and evaluating the new extension on established fit measures, uLDA is then tasked with approximating human-perceived topics. For this, a "ground truth" for topic labels is generated using a human-based experiment. These values are then used as a reference to be matched by the model output. Results show that the new extension outperforms traditional topic models regarding out-of-sample fit across all data sets and regarding human topic approximation for most data sets. These findings demonstrate that the novel extension can extract valuable information from the additional data conveyed by stop words and shows potential for better modeling natural language in the future.

Keywords: text mining, topic models, latent Dirichlet allocation, natural language processing, Bayesian models

Contents

List of Figures

List of Tables

List of Symbols

Dimensions

Symbol	*Iterator*	*Description*
D	d	number of documents in a given corpus
T	t	number of topics
W	i	number of words in the corpus
V	v	number of unique words in the corpus (*vocabulary*)
N_d	n	number of words in a document
S_d	s	number of sentences in document d
$M_{d,s}$	m	number of words in sentence s within document d

Note: Either of the indices (i), (d, n), or (d, s, m) can specify single elements (such as words) within a Corpus (dimension W).

Parameters

Symbol	*Dimension*	*Description*
C^{WT}	$V \times T$	a count matrix for the number of words per topic
C^{DT}	$D \times T$	a count matrix for the number of topic assignments per document
θ	$T \times D$	a matrix that consists of θ_d for all documents D
ϕ	$V \times T$	a matrix providing the probability for each unique word v within each topic t
ψ	V	a V-dimensional vector providing the probability for each unique word v within the non-topic
$\mathbf{w}$	W	a vector of word token

Parameters (cont.)

Symbol	*Dimension*	*Description*
$\mathbf{z}$	W	a vector of topic assignments
τ	W	a vector of binary classifiers for topic ($\tau = 1$) or non-topic ($\tau = 0$) assignment
w		a single word token
α	T	a T-dimensional vector providing the Dirichlet-prior for each θ_d
β	V	a V-dimensional vector providing the Dirichlet-prior for each ϕ_t
β_ϕ	V	a V-dimensional vector providing the Dirichlet-prior for ϕ
β_ψ	V	a V-dimensional vector providing the Dirichlet-prior for ψ
δ		corpus-wide a priori probability of topic-specific words $\mathrm{P}(\tau = 1)$
γ		a 2-tuple (γ_1, γ_2) providing the Beta-prior for δ

Part I

Introduction

Chapter 1

Motivation

When discussing statistical models for text data, a term that has gained more and more traction within the past years is natural language processing (NLP). It refers to the interaction between computer models and (usually) human-generated language. The expression "natural language" is derived from the fact that input data for these models take the form of unmodified human language. The definition is probably best illustrated by contrasting it to structured text data such as hashtags, tables, or even programming code.

However, the term is used somewhat loosely since most NLP models do not directly take natural language as input. Instead, the raw text data is pre-processed to make the data more compatible with model assumptions. Typical pre-processing steps include, for example, removing spelling mistakes, removing redundant punctuation, or even deleting punctuation altogether. Other common steps focus on removing words, such as frequent or infrequent terms or so-called stop words. Finally, some researchers also employ stemming, which removes grammar from words by reducing them to the word stem. This procedure is often followed by "quantifying" the text data through encoding terms into token vectors, which is mainly done for the sake of conforming unstructured language to structured models with limited capabilities.

These pre-processing steps present a common theme: removing structure and information from natural language is necessary to conform the data to model requirements. However, some key benefits of natural language are lost by these procedures since all the structure is part of what makes it so easily understandable to humans. Therefore, the question arises if all of the usual pre-processing steps are vital or if it would be

advisable to somewhat ease model restrictions in order to better harness this information that is otherwise lost.

Although there is plenty of research on this topic when looking at supervised models such as deep neural networks, these methods require labeled training data which provides additional hurdles for users. This thesis will instead focus on unsupervised models that can generate insights into previously unseen data without intensive training. One particular type of unsupervised models, namely topic models, will be at the heart of this work. These models are typically used to identify latent topics in both small and large corpora of text data.

Since resolving all the typical pre-processing steps would be an multifaceted task beyond the scope of one thesis, the attention of this dissertation is limited to the step of removing stop words. The following pages will examine if and how stop words can be introduced into topic models to avoid their removal and instead utilize the information conveyed by them. The main objective behind this approach is to find a better model for natural language.

However, defining what characterizes a "better" model is a challenge of its own, since natural language is not a phenomenon easily quantified. Especially when trying to evaluate topic models, established measures are more concerned with model fit or topic coherence than with how well the model represents language. As an approach to resolve this issue, this thesis also contains a novel experiment designed to obtain an approximation of natural language topics, which can then be used to evaluate topic models concerning their prediction for that data.

In order to solve these two tasks, some groundwork has to be laid first. The initial steps include re-defining the concept of stop words and their role in text data as well as descriptive analyses on several corpora to identify a suitable approach for including their information in topic models. Once the model is formulated, the experiment will provide human-generated topic labels that can then be used for evaluation.

Chapter 2

Contribution and Structure

The primary focus of this thesis is to examine the relationship between topics produced by topic models and topics as perceived by humans. This endeavor will entail examining the discrepancy between raw text data obtained as natural language and the pre-processed data fed into a topic model, with a particular focus on stop words. A novel model will be proposed that aims to handle data more closely to the original text. The model proposition is complemented by an examination of how humans do form topics when processing a corpus. The following questions formalize this objective:

1. Does removing stop words before model estimation change the interaction between LDA and the given data?

2. Can LDA be extended to harness possible interaction effects? Could this allow for a reduced pre-processing that would warrant the input data to be closer to natural language?

The natural next question would be whether the new extension can model human-perceived topics better than standard LDA. However, it this question cannot be answered without defining a human-based reference point first, which leads to these additional questions:

3. Can reference values for human-perceived topics be obtained in an experimental setting? How do humans perceive topics in the context of an experiment?

4. Does the proposed extension provide a demonstrably better model for human-perceived topics?

All these questions will addressed during the course of this thesis and picked up again in the conclusion part.

The presented work is comprised of four parts. In the first part, the *Introduction*, the motivation behind this thesis is described and the research objective is presented. Part II, *Theoretical Foundation*, introduces the primary reference model for this research, latent Dirichlet allocation, and provides an overview of the current state of research regarding topic model extensions concerned with relaxing related assumptions. It also contains the theoretical framework of *ubiquitous terms* suggested to replace stop words and motivates the model extension introduced later. The *Empirical Analysis* in part III is itself sectioned into four chapters. First, the data used in this study is presented in chapter 5, followed by a description of the employed methods used for answering the research questions in chapter 6. The results are presented in chapter 7 and discussed in detail in chapter 8. Part IV, *Conclusion*, provides a summary of the findings and offers possible future paths of research.

Part II

Theoretical Foundation

Chapter 3

Introduction to Topic Models

This chapter introduces topic models, specifically latent Dirichlet allocation as proposed for text data by Blei, Ng, and Jordan (2003). After laying the groundwork to foster understanding of topic models and their assumptions in section 3.1, section 3.2 will introduce the two main model restrictions this thesis aims to resolve. The first part examines model adaptations that aim to transfer some of the information on the internal structure of documents into the model to move away from the simplification of viewing documents as "bags of words." The second part is concerned with extensions to topic models that try to incorporate the information of stop words in some way.

Afterwards, chapter 4 will provide suggestions for addressing said limitations. First, existing research on how to approach the limitations are presented in section 4.1. Second, section 4.2 will introduce a novel approach to the issue of stop word information by introducing the concept of *ubiquitous terms*.

In general, topic models aim to identify unobserved, latent topics based on a statistical analysis of text data (Blei, Ng, and Jordan 2003; T. L. Griffiths and Mark Steyvers 2002; Hofmann 1999). The underlying assumption is that documents consist of a mixture of topics, i.e., that a document is merely the combination of terms that can be assigned to one of several topics (M. Steyvers and T. Griffiths 2010). Topics are, in this case, defined as a probability distribution over terms. Therefore, every word can be, in theory, assigned to every topic, but some assignments are much more likely than others. The topics are usually characterized

by their most likely terms. For example, a topic dealing with the optics of a product could include words such as "red," "blue," "color," "look," and "bright."

Topic models are so-called *generative models*, which means that they specify a probabilistic procedure that could be implemented to generate documents (M. Steyvers and T. Griffiths 2010). However, the actual use of topic models shows when the process is inverted by looking at given documents and estimating which model parameters are most likely to have produced these results.

The usage of topic models did not necessarily start with LDA. It was developed from other text mining approaches, such as latent semantic analysis (LSA), which used matrix decomposition to find the topics of a corpus. Still, LDA in the form proposed in the widely cited paper by Blei, Ng, and Jordan (2003), was arguably the most used topic model variant. To be precise, the *smoothed LDA* presented in the 2003 paper is nowadays considered "standard LDA" and will therefore be referred to as simply *LDA*. Section 3.1 will provide a closer look at that model, which will also function as a starting point for developing the ubiquitous terms model in section 6.2.

3.1 Latent Dirichlet Allocation

Latent Dirichlet allocation is a topic model that uses hierarchical Bayes analysis to make statistical inference about the latent topics. It was based on work about latent semantic analysis (LSA), frequently also referred to as latent semantic indexing (LSI) (Deerwester et al. 1990; Hofmann 1999).

Although the most prominent use case is text data, LDA and LDA-based model extensions such as mixed-membership models have been successfully used with a wide variety of data sets such as political voting data (Gormley and Murphy 2014) and text data (Grimmer 2010), population genetics (Shringarpure and Xing 2014), and image analyses (Cao et al. 2016). In the field of marketing specifically, there have been applications to mobile app usage data (Do and Gatica-Perez 2010), consumer reviews (e.g., Büschken and Allenby 2016), and purchase histories (e.g., Jacobs, Donkers, and Fok 2016; Ishigaki et al. 2015).

Model Proposal

Latent Dirichlet allocation inherits the main idea of topic models that documents can be represented by a collection of topics. A "topic" is defined as a probability vector across all terms of a corpus. As the model's name suggests, the terms within a topic are assumed to follow a Dirichlet distribution. The idea is that a corpus can be described with a set of T topics. In a marketing context, these could be different product aspects in a consumer review. For example, in the tent data set, reviews could describe the user experience with the tent in question by talking about its durability, look, or handling. However, not all reviews talk about all topics in the same scope. This is somewhat obvious since consumers can have different experiences with the same product or different preferences about its characteristics. Nevertheless, this variety in subjects is also a helpful circumstance when it comes to parameter estimation. The variance in topic shares is necessary to determine the topic probability vectors. If all documents had the same shares of topics, it would be impossible to estimate the topics due to a lack of information. The only observed variables in latent Dirichlet allocation are the document characteristics themselves. LDA assumes that the words within a document are distributed as

$$P(w_i) = \sum_{t=1}^{T} P(w_i \mid z_i = t) P(z_i = t) \qquad \forall i \in \{1, \ldots, W\} \,, \tag{3.1}$$

where T is the number of topics and W is the number of words in the respective document. Within the model, the probability of a word w given topic t is denoted as $\phi_t = P(w \mid z = t)$. The prior probability of the topic assignment z of that word being equal to topic t within document d is $\theta_d = P(z = t)$. A detailed overview of the notation used for LDA-type models within this dissertation is shown in the list of symbols preceding the thesis.

As mentioned before, topic models are considered generative models, and latent Dirichlet allocation is no exception. The generative process that would describe how LDA could produce documents is stochastic, which means that the results would be random. In theory, there is a chance that a document would be generated that matches exactly one of the reviews within a given data set. Some variables are considered fixed

a priori, such as the number of topics (T) and the vocabulary size across all documents (V). Additionally, α and β are positive vectors of length T and V, respectively. In the following description, $\text{Dir}_N(\gamma)$ denotes a Dirichlet distributed vector of length N with parameter γ, while $\text{Multi}(\delta)$ is a multinomial distributed value with probability vector δ.

Algorithm 1: Data generating process of LDA

for *each topic* t **do**
  Draw a word distribution $\phi_t \sim \text{Dir}_V(\beta)$
for *each document* d **do**
  Draw a vector of topic shares $\theta_d \sim \text{Dir}_T(\alpha)$
  for *each word* $w_{d,n}$ *in document* d **do**
    Draw the topic assignment $z_{d,n} \sim \text{Multi}(\theta_d)$, $z_{d,n} \in \{1, \ldots, T\}$
    Draw a term $w_{d,n} \sim \text{Multi}(\phi_{z_{d,n}})$, $w_{d,n} \in \{1, \ldots, V\}$

This process is visualized in a directed acyclic graph representation in figure 3.1. The observed random variable $w_{d,n}$ is denoted by the shaded node. The other, unshaded nodes, represent unobserved or hidden random variables. The notation also includes constant priors (in this case α and β), so-called *hyperpriors*. These are set a priori and can be used as tuning parameters. The edges denote dependencies between random variables and include the respective random distribution that shapes the relationship. The boxes follow the "plate notation" and denote replication.

Other than in classical mixture models (e.g., Nigam et al. 2000), in LDA, a document can inherit several topics. This characteristic is represented by the probability vectors θ_d and ϕ_t, which are themselves Dirichlet distributed. The Dirichlet distribution is predestined for this use as it produces positive vectors that sum to one, which is necessary for the subsequent draw from the multinomial distribution. The density of a Dirichlet distribution is

$$p(\theta \mid \alpha) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_i \theta_i^{\alpha_i - 1}, \tag{3.2}$$

where Γ denotes the Gamma function. If $\alpha_i = \alpha_j \forall i, j \in 1, \ldots, K$ with K as the length of vector α, the Dirichlet distribution is also called *symmetric Dirichlet*.

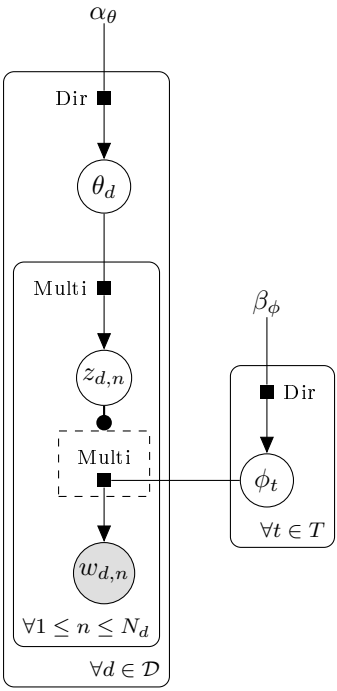


FIGURE 3.1: Graphical model representation of latent Dirichlet allocation.

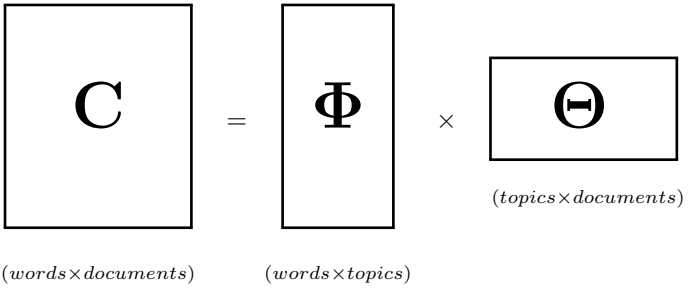


FIGURE 3.2: Visualization of the implicit matrix factorization.

Parameter Estimation

Mathematically speaking, LDA performs an implicit matrix factorization. A corpus of text is converted into a document-term-matrix, which tables the counts for each word within each document. The goal is to decompose this matrix into two matrices by introducing latent topics. One of the resulting matrices is the topic-term matrix, which contains the probabilities to observe each word within each topic, respectively. The other matrix is called the document-topic matrix and contains the probability that any observed word within a specific document is assigned to a specific topic. This process is visualized in Figure 3.2.

These matrices are also involved in the estimation process when using Gibbs sampling. Gibbs Sampling is a specific Markov chain Monte Carlo (MCMC) method, an iterative approach to sample values from complex distributions (Gilks, Richardson, and Spiegelhalter 1995). In Gibbs sampling, joint distributions of several variables are deconstructed into distributions of a variable subset, conditioned on each other's results. By sequentially sampling each distribution over and over again, the sampled values approximate the target distribution.

To further demonstrate the principles of Gibbs sampling, the implementation of latent Dirichlet allocation, as shown by M. Steyvers and T. Griffiths (2010), is explained. The notation introduced at the start of this work (page xv) applies here as well.

The goal of this sampling procedure is to approximate the posterior distribution of all terms as given by

$$P(w_i \mid \phi, \theta, \alpha, \beta). \tag{3.3}$$

Equation 3.3 can be divided up into distributions dependent on each other as follows:

$$P(w_i \mid \phi, \theta, \alpha, \beta) = P(w_i \mid z_i, \phi) P(z_i \mid \theta_d) P(\theta \mid \alpha) P(\phi \mid \beta) \tag{3.4}$$

Using this partition and keeping in mind that w_i is observed data, the Gibbs sampling steps can be identified as sampling z_i, θ, and ϕ, with α and β being constant priors that are treated as model parameters.

In this process, each word within each document is considered separately. For every word w_i, a topic z_i is sampled. Typically, LDA uses a collapsed Gibbs sampling step for this, which means that although z_i would be directly dependant on ϕ, that parameter is integrated out. Values of z_i are instead sampled conditionally on other values of z, w_i and the priors, which implicitly provide information on the distributions of topics and the terms within them. This sampling is done via:

$$P(z_i = t \mid z_{-i}, w_i, d_i, \cdot) \propto \frac{C^{WT}_{w_i j} + \beta}{\sum_{w=1}^{W} C^{WT}_{w,j} + W\beta} \frac{C^{DT}_{d_i t} + \alpha}{\sum_{u=1}^{T} C^{DT}_{d_i u} + T\alpha} \tag{3.5}$$

where $C^W T$ is a matrix with observation counts for all words within each topics and $C^D T$ is a matrix containing observation counts for all topic assignments within each document (see notation on page xv). Equation 3.5 can be used to produce an unnormalized probability vector of length T. This unnormalized probability is then divided by its sum to obtain the actual probability of $z_i = t$.

For estimation purposes, starting values of z_i are set randomly for all tokens w_i within the corpus. From this, C^{WT} and C^{DT} can be calculated and then used to generate draws of z_i. Since every draw of z_i can only be dependent on every *other* z_i (as depicted in equation 3.5 by z_{-i}), the matrices cannot be used as-is for each draw. Before each sampling step, both matrices are decremented by 1 at the position corresponding to w_i, d_i, and z_i. After a new z_i^* is generated, the matrices are incremented by 1 for w_i, d_i, z_i^*.

Since equation 3.5 does not contain ϕ nor θ, technically, it would not be necessary to sample either one of those hidden variables. However, since both values are used for inference, it is advisable to sample them during the Gibbs process to obtain their respective posterior distribution. Moreover, they can be used to calculate the model log-likelihood to track convergence.

The sampling of ϕ and θ is done via a simple Dirichlet draw using the current count as distribution parameters. The expressions are:

$$\phi_t = Dir(C_{.t}^{WT} + \beta) \qquad \forall t \in \{1, \ldots, T\} \tag{3.6}$$

and

$$\theta_d = Dir(C_{d.}^{DT} + \alpha) \qquad \forall d \in \{1, \ldots, D\} \; . \tag{3.7}$$

After ϕ and θ are sampled, the whole process starts over. At the start, there is a so-called burn-in period in which the process is still dependent on its starting values and does not yet generate draws from the posterior distribution of z (M. Steyvers and T. Griffiths 2010). After this period, the generated values represent a good approximation of the target posterior distribution.

It should be noted that the topics are ordered randomly dependent on starting values only. This means, for example, if there is a topic about "tent set-up," it should appear in every Gibbs sampling result if it is a stable topic. However, at which point t ($t \in \{1, \ldots, T\}$) this topic appears is inconsistent across model instances, which must be kept in mind when different runs are compared.

One of the disadvantages of the original LDA approach is that the number of topics is externally fixed, which means that T is another parameter to be optimized. Luckily, there already have been a variety of approaches to solving this problem in the literature. Techniques range from simply estimating several models with different topic numbers and compare them by a certain measure, such as model log-likelihood or topic perplexity (e.g., Blei, Ng, and Jordan 2003), to more complex solutions incorporating the topic number selection process within the Bayesian model itself (e.g., Blei, T. L. Griffiths, et al. 2004; Teh et al. 2006). However, the task of setting the optimal number of topics is not a focus of this work. Instead, sensible values are used and the main findings will be tested to be robust against changes in T.

Model Results

Assuming all parameter tuning was done correctly, the question arises of what to do with the results. The first step might be to visualize the results to the user to give an overview of the model's expressiveness.

The first possible step is to visualize the resulting topics. This is usually done by looking at the most frequent terms of that respective topic, i.e., those with the highest values in the probability vector $\widehat{\phi}_t$ for topic t. Table 3.1 shows this for some LDA results using the *Brexit* data set. Blei and Lafferty (2009) proposed a different ranking method that gives more importance to terms that are predominant in one topic only by using the following equation:

$$\text{term-score}_{t,v} = \widehat{\phi}_{t,v} \cdot \log \left(\frac{\widehat{\phi}_{t,v}}{\left(\prod_{j=1}^{T} \widehat{\phi}_{j,v} \right)^{\frac{1}{T}}} \right) \tag{3.8}$$

with $t \in \{1, \ldots, T\}$ being the respective topic and $v \in \{1, \ldots, V\}$ denoting the respective term in the vocabulary. Table 3.2 again shows some top words of an LDA over the *Brexit* data set[1] using their approach. Although there are only a few changes in the word order, some of the top words differ.

These terms are usually used to identify the primary concern of each topic. For a human, there might be an intuitive answer on how those topics could be described. For example, looking at Topic 10 from tables 3.1 and 3.2, a person somewhat familiar with the topics of the Brexit discussion could recognize it as the Northern Irish backstop issue, which was discussed extensively at the time. However, not every topic is as clearly defined, and while looking only at the top words makes it easier for humans to grasp the "topic" of a word collection quickly, it could be argued that a large part of the topic is ignored since vocabularies often range in the thousands. However, automatic topic labeling methods are not yet well established. Some promising work on this was done by Mei, Shen, and Zhai (2007). Since this is not an essential focus of this thesis, the model analysis will forego labeling the topic and will use simple ϕ values for ranking terms.

[1] The data sets used in this thesis are described in section 5.1.

TABLE 3.1: The top 10 words ranked by $\widehat{\phi}$ for some of the topics from an LDA on the *Brexit* data set (in descending order).

Topic 3	*Topic 4*	*Topic 8*	*Topic 10*
johnson	was	labour	ireland
deal	remainers	party	eu
will	you	election	northern
eu	about	voters	the
he	leavers	vote	border
mr	his	corbyn	uk
trade	bbc	are	deal
uk	gove	remain	would
his	her	they	agreement
free	history	tories	irish

TABLE 3.2: The top 10 words of some of the topics from an LDA on the *Brexit* data set ranked by a term-score according to Blei and Lafferty (2009) (in descending order).

Topic 3	*Topic 4*	*Topic 8*	*Topic 10*
mr	remainers	labour	ireland
johnson	gove	party	northern
deal	bbc	election	eu
trade	you	voters	border
will	leavers	corbyn	the
eu	about	vote	uk
uk	was	are	irish
he	media	remain	agreement
his	her	farage	deal
market	his	lib	would

3.2 Limitations

Although LDA shows good performance for various uses, it is still subject to certain assumptions and restrictions. Disregarding the distribution assumptions, which were already discussed in the previous section, there are two issues worth pointing out. First, the model operates under a *bag-of-words within documents*, often abbreviated as *BoW*, assumption. This premise leads to the order of terms within each document being ignored by LDA. Second, the model relies on pre-processing, which prepares the data for the estimation routine and tends to eliminate the larger share of terms within a corpus beforehand.

Bags of Words

The *bags of words* assumption is a widely used phrase to describe that a model such as LDA has no concept for word order, and therefore the actual sequence within a document does not impact the model results. The image is that all words of a document could be put in a bag and shaken, completely randomizing the order while still producing the identical posterior distribution according to LDA.

This assumption follows directly from the generative process. When generating a document d with LDA, every word within the document is an i.i.d. (independent and identically distributed) draw $w_i \sim \text{Multi}(\phi_{z_i})$ following an i.i.d. draw for $z_i \sim \text{Multi}(\theta_d)$ (in standard Gibbs sampling). Therefore, the probability for each word does not change depending on its position inside the text. However, this assumption has some implications that will be examined in the following paragraphs.

On the positive side, it helps with the variety of speech. When looking at short windows such as individual sentences, the order of words can often be changed while still conveying the same meaning. For example, "We were able to set up the tent in just five minutes." and "In just five minutes, we were able to set up the tent." basically provide the same information, with a slightly changed word order. LDA treats both the same, making it easier to compare different people talking about the set-up process and correctly group them into one topic. By ignoring the structure within a document, it is also easy to identify topics spread out across a document, for example, if a review contains the topic of "material quality" on several paragraphs, separated by different topics in between.

However, this simplification also ignores a lot of the information which can be conveyed by language structure. It stands to reason that authors do not produce a text by randomly changing the topic after every word. It seems more logical that words within proximity of each other are more likely to be of the same topic than words separated by several sentences. Incorporating that information in a model could improve topic estimation and, therefore, overall model performance. Section 4.1 will give an overview of existing research on this topic.

Not Using All Available Data

The second limitation that will be spotlighted here is the pre-processing of data, especially the step of removing so-called "stop words." When discussing implementations of LDA, the input text is usually converted into *(word) tokens*. These are generated by splitting documents at whitespace characters into lists of words (or sometimes numbers, which is why the term *token* is probably more accurate). However, in reality, the transformation from raw text into token lists is not quite as simple. Unprocessed text is not only filled with words but also with numbers, punctuation marks, and other special characters such as hyphens or quotation marks. The usual approach to pre-processing for LDA throws out everything that is not a word (or number). The subsequent step can then be to remove additional words, for instance, *rare terms* and *stop words*.

The expression "rare terms" describes tokens that appear very infrequently within the corpus, with the definition of "infrequent" varying between studies. Nevertheless, the motivation behind removing those terms can be explained most easily when looking at the extreme case of a token with one single occurrence. A term that only appears once has a single co-occurrence with every term within its document. Determining which topic within this document the term belongs to is impossible without additional information – a single data point cannot form a pattern. The draw would be solely depending on the prior. By the same logic, rare terms cannot inform the model on other terms' topics since the co-occurrence is too infrequent. Every additional occurrence increases the confidence with which the related topic(s) can be determined. Nevertheless, even if the rare terms were assigned to one topic, there would be barely any benefit. The terms could never be important enough for a

topic (with importance measured by frequency within this topic) to gain helpful insight.

All that being said, the question could arise why even bother removing terms that would probably not show up in the resulting visualizations. The main reason behind this is simply performance. Removing rare terms shrinks the vocabulary V by a surprisingly large amount. In an exemplary data set, the share of single occurrence words can be up to roughly 50%.[2] A smaller V reduces the dimensionality of C^{WT} and the total number of tokens W in a corpus, which reduces processing time for Gibbs sampling in two ways. First, the number of words directly impacts the number of draws for z_i, which reduces runtime. This effect is arguably relatively small since single occurrence words also count only for one total occurrence in the corpus. Second, when the dimension of C^{WT} is reduced, the evaluation of equation 3.5 is faster as well.

The other type of removed terms, *stop words*, can not easily be dismissed since they are usually frequent enough to play a role in the topic formation process. The term "stop words" describes words that do not bear much meaning and are instead helping out by providing syntax. Typical examples of this are function words such as "and," "the," or conjugations of "to be." Frequently, these terms are removed via stop word lists. In that case, all terms provided by a given stop word list are eliminated from the corpus. Among the most prominent ones of these lists are the NLTK[3] and the *snowball*[4] stop word list, which also contain words as "very," "too," and "not," which might arguably transport meaning in specific contexts. Stop words usually appear numerously all across the corpus.

One of the main points of this thesis is to demonstrate that these terms should not be discarded beforehand since they can still provide information to the model. Section 4.2 will go into detail and explain why it is a reasonable assumption that stop words are relevant for topic models. Therefore, it introduces the concept of "ubiquitous terms" to better determine the contribution of those terms to a model. In the meantime, section 4.1 provides an overview of existing research on incorporating the information provided by such terms into topic models.

[2] For example, the Brexit data set presented in section 5.1 has about 47% single occurrence words

[3] The Natural Language Toolkit is a widely-used open-source package to facilitate language modeling in python

[4] Snowball is an open-source project which aims to facilitate stemming for text mining

Chapter 4

Addressing LDA Limitations

This chapter will provide a variety of ways to approach the limitations presented in section 3.2. First, section 4.1 will outline existing approaches on both bags-of-words and stop words. Section 4.2 will then provide a novel perspective on stop words and their relation to topics, which will constitute the theoretical framework upon which the new model will be formulated in section 6.2.

4.1 Existing Approaches in Literature

Bag-of-Words within Documents

A wide range of publications already addressed the issue of including the document structure into a topic model in one form or the other. Wallach (2006) approached this problem by introducing a word-to-word dependency, making the draw of w_i dependant on z_i as well as w_{i-1}. Büschken and Allenby (2016) introduced a constraint that forced words within one sentence to be assigned the same topic, thus creating SC-LDA (sentence-constraint latent Dirichlet allocation). A similar approach was chosen by Gruber, Rosen-Zvi, and Weiss (2007), who postulate that topics are identical for certain blocks within documents and only change at specific points dependent on a change probability. Moody (2016), on the other hand, combined the Dirichlet-distributed word spaces from LDA with the skip-gram methodology as implemented by *word2vec* to

create *lda2vec*. Shafiei and Muios (2006) created latent Dirichlet co-clustering, in which they assume that the bag-of-words assumption does not hold for the whole document but only within segments of it, e.g., individual paragraphs. In this case, every segment can be assigned a different θ. Other approaches use external data to enrich the bag-of-words representation, leading to non-i.i.d. draws of z. Examples include defining keywords (Ramage, Dumais, and Liebling 2010) or tags (Yang et al. 2015).

In the scope of this thesis, incorporating language structure into the topic model will rely on the existing research of Büschken and Allenby (2016). This will be picked up again in section 6.2.2, where their SC-LDA model will be adapted to incorporate some form of structure into the topic model proposed in section 6.2.1.

Using Stop Word Information

The issue of harnessing information of otherwise removed words has been addressed from several angles in the literature. T. L. Griffiths, Mark Steyvers, et al. (2004) combined LDA with a Hidden Markov Model (HMM) to classify terms in different syntactic groups, with only one group containing topic words. Although this model provided additional information by including some sort of part-of-speech-tagging, it was outperformed by LDA in document classification tasks. Schofield, Magnusson, and Mimno (2017) experimented with removing stop words before and after LDA estimation. They showed that both methods have advantages and disadvantages, dependent on the measure of quality, with a tendency to improved coherence when estimating the model before removing stop words. However, none of their approaches included stop words in the evaluation period. Wallach, Mimno, and Mccallum (2009) did not remove stop words but included an asymmetric prior over θ, to the effect that stop words were grouped into one topic. This method is effectively a similar approach to the model proposed in section 6.2.1 but leaves more uncertainty to the symmetry of priors between actual topics since these are also left to the model estimation process. Dolamic and Savoy (2010) did not apply LDA but showed that the selection of stop words matters for different information retrieval methods. However, stop word lists vary greatly and are mainly based on the researchers' opinion (Fox 1989), which means that their findings point to the fact that opinion possibly can influence model performance. With the goal of tempering

this influence, a more automated approach will be put forward for the modep proposed by this thesis..

Another point worth mentioning is that besides research on topic models, language studies have shown that the use of pronouns (which are typically among the removed stop words) was shown to be related to both self-perception (Pennebaker 2011) and perception of the writer (Packard, Moore, and McFerran 2018). Including these terms in the data therefore can arguably convey valuable information, especially in a marketing context.

This thesis focuses on a simple, automated approach that does not require additional input by the user and can easily be incorporated into existing LDA extensions. In the next section, a theoretical framework for the distribution and information of "stop words" and similar terms within data is developed and tested. Section 6.2 will then propose a new extension to LDA based on those findings.

4.2 Introducing Ubiquitous Terms

This section will focus on moving topic model input closer to processing natural language by addressing the limitations expounded in section 3.2. As mentioned earlier, the issue of grammatical structure will be left to the existing research of Büschken and Allenby (2016). However, the information loss due to stop word removal will be approached from an entirely new angle. The novel concept of ubiquitous terms is designed to differentiate between typical stop word lists and "true" non-topic terms.

The following pages will now introduce the theoretical groundwork upon the rest of this thesis is built. This includes presenting the concept of ubiquitous terms and explaining how it differs from typical stop word lists. This theoretical framework will later be reinforced by a descriptive analysis of the data sets in section 5.3.

When talking about stop words, the emphasis is on terms such as function words, which are perceived as "meaningless" when talking about topic modeling. Stop words are usually compiled by consulting stop word lists, but sometimes include high-frequency words as well. It is assumed that they do not contribute to topic models significantly since they occur independently of the underlying topic. For now, this claim is not contested for argument's sake, although it will be examined in detail in section 5.3. The assumption is that words such as "is" and "the"

appear in every context with similar frequency. Thus, these terms would not give additional information about the co-occurrence of terms since the occurrence has a too low variance. Since the terms are usually rather prevalent, they would be expected to show up in topic-related top words across all topics.

The main postulation of this thesis is that the established definition and handling of "stop words" should be considered obsolete. First, determining stop words by a pre-defined list can be arbitrary. Although stop word lists usually have some foundation in research, there still is a wide variety of lists that sometimes also get adapted by researchers, which can lead to a bias introduced by the removal. Second, removing these terms is thinning the data available to the model. There is a case to be made that specific terms that usually appear on such lists might get used more frequently in specific contexts – i.e., specific topics – than others.

The word "to" can serve as an example of this since it is typically included in stop word lists. In many circumstances, the term is used as a particle, e.g., "I tried to set up the tent" or "I bought the toy to surprise my dog." There is arguably not much meaning in this word within that context. However, the word can be used with a variety of other meanings as well. For example, it is used in a temporal context in a sentence such as "setting up the tent took 10 to 15 minutes" or "the shop is open from 9 to 5." In this case, the context could arguably be topic-specific. Another possible utilization is in a spatial context. Examples would be "it is 400 meters from the hotel to the beach" or "you can get to the city center in only a few minutes." A result of this could be that the term "to" is more likely in the context of time frames or relative spatial positioning than it is in other contexts. Hence it might be more likely to appear in specific topics than in others, which contradicts the stop word assumption that the terms are irrelevant – or at least agnostic – to topic content.

However, this is most certainly not true for all stop words. The best example for a term that would probably be used frequently regardless of the context is the definite article "the." This word is such an elementary part of the English language that it would most likely give no information about the underlying topic to a model. The argument for removing it, therefore, has some merit.

At this point, the concept of "ubiquitous terms" is introduced to better distinguish between "stop words" as defined, for example, by stop word lists and terms such as "the," which are "true" stop words in the

original sense. Ubiquitous terms are, as the name indicates, rather omnipresent in a constant frequency, with the latter part being the crucial difference. Their frequency comes with such a low variance that it could be considered constant across documents. Since topic models describe documents as entirely comprised by varying compositions of latent topics, it follows that the frequency should be near-constant across topics as well.

In contrast to the straightforward definition that stop word lists provide, categorizing whether or not a word counts as a "ubiquitous term" does not have to be binary. To recall the example made earlier, the term "to" can be both considered ubiquitous and topic-specific. Since it is among the most frequent words according to the *Corpus of Contemporary American English* (Davies 2008), it would appear in high frequency across all documents. However, as explained above, it might also occur a little more frequently in specific contexts. It might depend on the corpus whether or not the variation in use can be assigned to a topic. Additionally, the determination of ubiquitous terms is agnostic to the respective grammatical form, which means every part of speech could be a ubiquitous term, and none has to be.

In conclusion, this work will employ the novel concept of ubiquitous terms in order to identify non-topic words in a flexible, stochastic way while still making use of their information. This approach aims to replace the existing procedure of consulting fixed stop lists that relies on simply discarding a pre-defined set of words and thus losing all related data.

Part III

Empirical Analysis

Chapter 5

Data

With the theoretical foundation all set, the upcoming chapter will provide information on the data used for the subsequent analyses. After presenting the data sets in section 5.1, the pre-processing procedure is described (5.2). An extensive descriptive analysis follows in section 5.3, where term frequencies and their distribution within the three corpora are examined.

5.1 Data Sets

This section will examine the data that is used for all following analyses. After a summary of descriptive statistics and meta-information about the data sets in use, the data preparation methods are discussed.

The analysis is based on three main data sets. Two of those consist of customer reviews, one regarding a camping tent and one regarding a pet supply store. The third data set consists of editorials from different news outlets that are concerned with Brexit. The latter is used as an example for more complex texts, as the length and complexity of documents are higher than for average online reviews.

Camping Tent Data Set

The camping tent data set is a collection of Amazon reviews for a product line of a roomy tent. It was chosen because it provided a large number of reviews that had mixed ratings. Although the average rating is about 4.2 on a 1 to 5 scale, there are still over 1000 reviews rated 2 stars or lower. This distribution indicates that the topics these reviews talk

about might be quite diverse. The raw set contains 7976 reviews and will be referred to as "tent."

Pet Supply Data Set

The pet supply data set is a collection of online reviews for a range of pet supply products. These products mainly consist of dog or cat food and other food supplements for pets that claim to provide health benefits. The ratings are a bit more leaning towards five stars than in the tent data set. The average rating is at 4.4, with about 670 reviews scoring at or below 2 stars. The unprepared set contains 6053 reviews and will be referred to as "dogfood."

Brexit Editorial Data Set

The Brexit data set contains editorials that were gathered from different online news outlets. The main criterion was that the article had to be about Brexit and that it had to be a long-form opinion piece. This lead to the documents being longer and more complex than in the other two data sets. However, since this is not a review data set, no ratings can be examined. This set contains 168 editorials and will be referred to as "Brexit."

5.2 Data Preparation

All data sets were put through the same data preparation protocol. The amount of pre-processing differs only for the separate analysis steps, which will be laid out here.

5.2.1 Preparation for the Descriptive Analysis

For the descriptive analysis, the data sets were put through minimal preparation procedures. Since the point of this thesis is to examine text at a level that is as close to natural language as possible, the pre-processing was very limited. In the first step, duplicates were removed. The second step was to de-capitalize all text and clean it from unwanted symbols and non-printable characters while also trying to fix encoding

issues that can occur when loading text data containing emojis or non-Latin characters. Apostrophes were removed while trying to concatenate negative clauses such as "don't" or "didn't" into single tokens (i.e., "dont"). After the text was cleaned, it was broken into words (tokens) by splitting the text at each whitespace character. After the pre-processing, each document is represented by a token vector, each token representing one word.

5.2.2 Preparation for the Model Estimation

Since the data would be used for sentence constraint topic models down the road, punctuation marks were used to identify sentence boundaries. Sentences are set to end on full stops as well as exclamation and question marks. The identifying characters themselves were not kept. The words were counted, and some tokens were removed, namely so-called "short tokens" that are not actual words but rather letters separated due to various issues (such as spelling or encoding errors). Moreover, tokens that appeared in less than three documents within the corpus were removed as well. Therefore, depending on the intended model, documents are represented by either a vector of tokens or a list of token vectors for bag-of-words or sentence constrained models, respectively. Table 5.1 provides an overview of some descriptive statistics after pre-processing. Note that although the Brexit data set has very few documents compared to the other two data sets, it still has a reasonable amount of total tokens (i.e., words) since the average document is more than ten times as long.

TABLE 5.1: Descriptive statistics of the data sets used.

				document length		
data set	*#docs*	*#tokens*	*vocabulary*	*mean*	*median*	*sd*
Brexit	167	115'961	4'073	694.38	724	409.78
dogfood	5'795	214'181	3'202	36.96	22	51.91
tent	7'851	697'065	5'462	88.79	52	115.43

5.2.3 Preparation for the Human-Based Experiment

The experiment is conducted on subsets of all three data sets. Table 5.2 shows the subset size in comparison to the complete data sets. Since the experiment relies on human processing, the amount of text made it unfeasible to process the entire data sets during the experiment, with the *Brexit* data set being the only exception.

TABLE 5.2: Descriptive statistics of the human-labeled data.

data set	*tent*	*dogfood*	*Brexit*
total number of documents	7'851	5'795	167
number of documents in subset	500	982	167
subset share of total size	6.4%	16.9%	100%

5.3 Extended Descriptive Analysis

Following these short statistics on the given corpora, an extended descriptive analysis is conducted to better understand the data, especially concerning ubiquitous terms as proposed in section 4.2. This exploration will underpin the concept of "ubiquitous terms" and point out how to approach the issue when extending LDA later in section 6.2.1. While the first part is a purely model-free frequency analysis, the second approach will look closer at the interaction between data and LDA when stop words are present. Based on these findings, section 6.2 will present an extension to LDA that aims to incorporate ubiquitous terms instead of deleting stop words along with all information conveyed by them. The data examination begins with a frequency analysis of words within documents. Its goal is to improve the understanding of how words are distributed in documents, with a particular focus on stop words to sharpen the distinction between *stop words* in the traditional sense and *ubiquitous terms*.

5.3.1 Model Free Frequency Analysis

This section aims to quantify ubiquity within the scope of the presented data. The task will be approached by studying word frequencies on a document level. To further examine the data with regards to ubiquitous terms, an in-depth analysis of the distributional characteristics of words is conducted to identify *ubiquitous terms* and describe how they relate to stop words. The results will show that a discrete allocation of terms to stop words is not advisable since there is a smooth transition of dispersion measures from "ubiquitous" terms to "normal" terms. Stop words and *ubiquitous terms* are not identical but overlapping. It is expected that most ubiquitous terms would appear on typical stop word lists.

In preparation for the following analyses, the first step is to count the occurrences of each word within each document, resulting in what is typically referred to as a "document-term matrix," which will here be denoted as C^{DW}:

$$\mathbf{1}_w(w_i, w^*) = \begin{cases} 1, \text{if } w_i = w^* \\ 0, \text{otherwise} \end{cases} \tag{5.1}$$

$$C^{DW}_{d,w^*} = \sum_{i=1}^{N_d} \mathbf{1}_w(w_i, w^*) \tag{5.2}$$

By dividing this count matrix by the total number of terms in the respective document N_d, a document-term-frequency matrix F^{DW} can be calculated in which each term gets assigned its share across words within each document using

$$F^{DW}_{d,w^*} = \frac{C^{DW}_{d,w^*}}{N_d} = \frac{\sum_{i=1}^{N_d} \mathbf{1}_w(w_i, w^*)}{N_d}. \tag{5.3}$$

In a first step, the most frequent terms as defined by the absolute counts within the corpus were inspected. As table 5.3 shows, the ranking slightly differs from those when looking at the average within-document frequency. For example, in the *tent* data set, the term "up" is not among the most frequent terms overall but appears to be frequent in many documents. This pattern might occur because the setup process is part of most reviews, but the description does not require extensive use of the term. Going forward, the focus of this work will be on the latter definition that refers to the average within-document frequency. This

TABLE 5.3: The most frequent terms overall vs. highest average frequency across documents for all three data sets.

tent		dogfood		Brexit	
overall	by doc.	overall	by doc.	overall	by doc.
the	the	the	my	the	the
and	tent	and	dog	to	to
it	and	i	the	of	of
tent	it	to	it	a	a
to	to	a	and	and	and
a	a	my	food	in	in
i	i	dog	dogs	that	that
in	up	it	this	is	is
of	this	food	i	it	for
this	for	this	to	for	it

Note: *Overall* refers to the frequency of a term within the whole corpus. *By doc.* refers to the mean of all within-document frequencies of a term.

interpretation is more suitable, given that the definition of *ubiquitous terms* is based on their appearance across all documents.

It could be tempting to simply use the frequency standard deviation to evaluate the dispersion of word frequencies. However, this measure would be biased since it prefers very infrequent terms, which is evident when recalling the formula for the standard deviation,

$$s = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}. \tag{5.4}$$

For rare terms, $\bar{x}$ and all x_i are very small, which leads to a small s. For very frequent words, $\bar{x}$ is large, and even relatively small deviations are, in absolute terms, much higher than they can ever be for rare terms. Using an actual numbers example, when the highest frequency of a term r is $x_r = 0.0001$, then the standard deviation cannot be higher than that. This effect is amplified by the fact that the values cannot fall

below 0. If the highest frequency for a frequent term f is $x_f = 0.1$, then the standard deviation can easily assume values above x_r even if the changes are small relative to 0.1, e.g., only in the range $[0.09, 0.11]$. This whole issue stems from the standard deviation being a non-normalized measure.

There are different dimensionless measures available to choose from, but not all are equally suitable. For example, the *quartile coefficient of dispersion* is defined as

$$QCD = \frac{Q_3 - Q_1}{Q_3 + Q_1}, \tag{5.5}$$

with Q_1 and Q_3 being the first and the third quartile, respectively. However, this has the disadvantage of not being defined for values with $Q_1 = Q_3$. Unfortunately, this happens to be most of the terms for short documents such as customer reviews, where $Q_1 = Q_3 = 0$.

Instead, this study employs the more suitable *relative standard deviation*, also known as the *coefficient of variation*. It is commonly used for comparing the variation of samples that have different scales (Van Valen 2005). This measure is defined as the quotient of the standard deviation and the arithmetic mean of the data, ergo:

$$RSD = \frac{\sigma}{\mu} \tag{5.6}$$

This measure is also closely related to the index of dispersion $\frac{\sigma^2}{\mu}$, with the only difference being the use of σ^2 instead of σ. Since this would not change the ranking of terms, both could be used interchangeably within the scope of this analysis.

Although the RSD can be biased on smaller sample sizes, it is still the most promising measure for this task. Since the analysis is based on the same sample size for all terms, the number of documents, any bias would be applied equally. Moreover, since it is only used to compare values between terms, a bias in the absolute value would have no relevant impact.

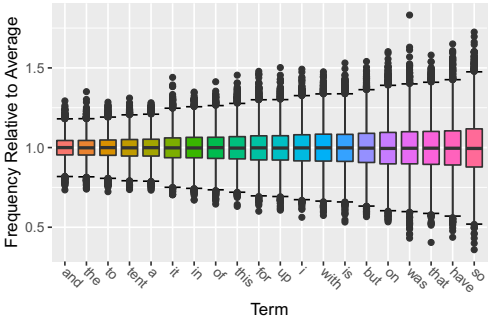
Another challenge of this analysis is that it is very sensitive to short documents since documents with only a few words might lead to the frequency of some words shoot up to values above 20 percent, which is not representative of the actual use within the English language. In reality, consumer reviews can indeed be very short, which could potentially

distort the results. To counter this issue, a proxy data set was compiled by sampling 1% of the documents randomly and combining them into a new virtual document. This step was repeated 1000 times to generate the new corpus. Thus, very short reviews cannot distort the data as much since several documents are combined. These results were checked for robustness by generating all of the following plots with the unaltered data sets as well, provided in Appendix A. Although the dispersion tends to be higher, the key results remain unchanged. Note that this simulation was not applied to the *Brexit* data set since it consists of fewer, longer documents that are more representative by themselves. However, for the sake of completeness, the simulation results are included in the appendix as well.

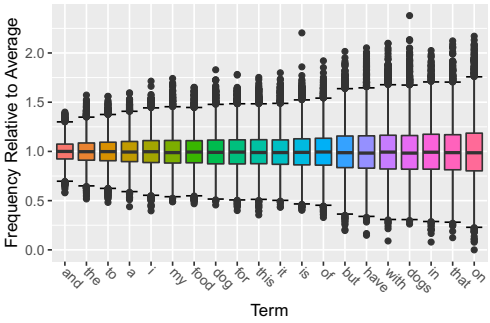
With the method set, the *ubiquitous terms* are examined further by sorting the vocabulary by the relative standard deviation in increasing order. For the top 20 terms, i.e., the terms with the lowest RSD, box plots are created showing relative deviation from the mean frequency. The values are obtained by dividing the term frequencies across all (virtual) documents by their mean. Figure 5.1 displays the box plots for those terms with the lowest relative dispersion across documents.

As was expected, there are many "typical" stop words in that list, such as "the," "and," and "a." However, the list also includes words that are widely used in that specific context, for example, "tent," "food," or "Brexit." Naturally, observing the term "tent" in a data set of tent reviews is not surprising. Some of the high-ranked words seem to be stop words but can be framed in a topic-specific context. When looking at the *tent* data set, the term "up" is most likely related to the process of setting up the tent. The *dogfood* data set contains frequent mentions of the reviewer's dog or dogs, which is represented by terms such as "I," "have," and "my."

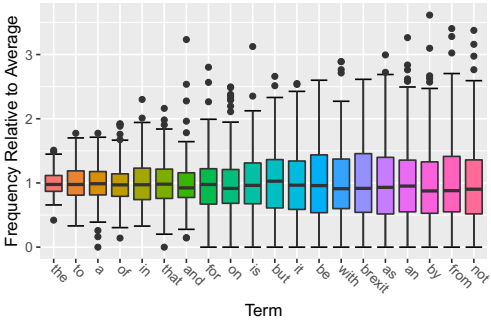
Another noteworthy piece of information is that the most ubiquitous words in the consumer review data sets are similar, with terms such as "I," "it," "a," "and," and "to" among the leftmost words. However, the word "I" does not even appear for the *Brexit* data set, and "it" is less frequent. This might be an indicator that the language used in both cases is different. Especially with the pronoun "I," this makes sense since editorials are rarely written in the first person, while consumer reviews use this perspective very frequently when writing about the experience with the product.



(A) Tents



(B) Dogfood



(C) Brexit

FIGURE 5.1: Frequency distributions for the different data sets.

Table 5.4: List of ten stop words used for the LDA-based exploratory analysis.

Stop word
the
and
to
a
it
I
of
in
this
is

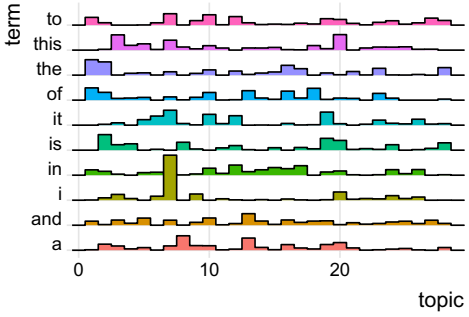
In summary, the first analyses strongly support the concept of ubiquitous terms. For the next step, section 5.3.2 will use latent Dirichlet allocation to further examine the characteristics of stop words and ubiquitous terms in the context of topic models. Afterward, section 5.3.3 will provide a summary as well as a brief discussion of the results.

5.3.2 LDA-Based Exploratory Analysis

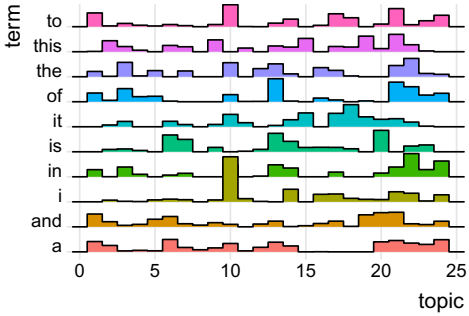
The upcoming paragraphs will further examine the data concerning its stop words in the context of an LDA model. This will help to gain a deeper understanding of what role stop words play in an LDA's topic and word distribution. For this, an LDA model is estimated using the described data sets and typical hyperprior settings[1]. For the following analysis, the posterior distributions of ϕ, θ, and z are used.

The first step is to focus on a specific set of common stop words as they can be found in stop word lists. Since those lists can get rather long (Dolamic and Savoy 2010), a subset of stop words was defined. For this, all data sets were pooled, and word counts were calculated. The subset was defined as the ten most frequent stop words in the pooled data set. The list is displayed in table 5.4.

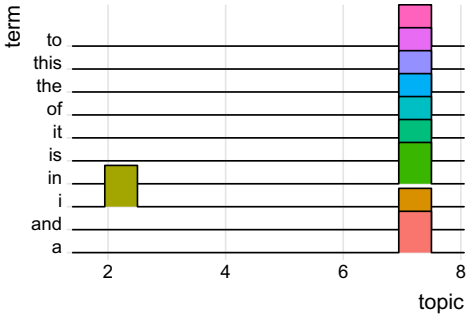
[1] To be precise: $\alpha = 0.1$, $\beta = 0.1$, $T = 20$



(A) Word distribution for the *tent* data set



(B) Word distribution for the *dogfood* data set



(C) Word distribution for the *Brexit* data set

FIGURE 5.2: Distribution of selected stop words amongst topics.

FIGURE 5.3: Word cloud of the peculiar topic in *Brexit* LDA.

Following the idea that stop words are occurring without relation to topics, these stop word terms should be approximately equally distributed amongst all topics. However, as the ridgeline bar plots in figure 5.2 show, that does not seem to be the case. These figures display the frequency of each word across all topics. It can be seen here that stop words are not distributed equally among topics and that there are remarkable differences between topics among certain words, which indicates that these stop words are used topic-specific and should be incorporated into a model instead of being removed from the corpus. The graphs show that some topics generally contain more stop words than others, while certain stop words can be almost entirely clustered into different topics entirely, such as the pronoun "I." This indicates that some topics are more related to personal experience, which would be expressed in the first person, than others.

Figure 5.2 also shows a peculiar result for the *Brexit* data set. Almost all stop words seem to be grouped within one topic, with the term "I" being the only exception. This anomaly hints at an interesting model behavior. As is confirmed by Figure 5.3, that particular topic contains terms unrelated to "actual" topics predominantly. Since this phenomenon is closely related to *ubiquitous terms*, Section 6.1 of the methods chapter will pick it up again in a short analysis.

5.3.3 Summary

The results of these two analyses indicate that stop words seem to be partially identifiable just by looking at the data without actually processing the word meaning and without applying any model. However, it is still possible for some of those terms to be topic-related depending on the circumstances of use, and other typical stop words do not appear here. Nevertheless, there is no clear threshold by which words can be grouped into "stop words" or "topic words." The transition is smooth and does not indicate that a natural border exists, which is taken as supporting evidence for the claim that distinct assignment of "topic" and "non-topic" for terms is inadvisable, and defining a cut-off value for topic words would be the same arbitrary definition that was criticized earlier. Some terms are most likely non-topic related, while others could doubtlessly appear in both topic and non-topic contexts. This ambiguity will be reflected in the proposed model in chapter 6.2 by a stochastic assignment.

In conclusion, the results show that frequency dispersion is a good indicator when examining *ubiquitous terms* and stop words. However, such measures should not be employed to define rigid stop word lists since the topic-relation can be ambiguous. Nevertheless, when integrated into a stochastic process, these model-independent frequency dispersion measures can be utilized to extend existing topic models such as LDA.

The results of the LDA-based analysis re-emphasize the fact that expanding LDA to include ubiquitous terms in the model process without removing stop words is a consequential next step. Section 6.2 will later combine all previous findings in the proposal of a novel extension to LDA that incorporates these terms with new variables in the model.

Chapter 6

Methods

The foundation built in the previous chapters is now used to improve LDA by incorporating the concept of ubiquitous terms. In the pages leading up to this chapter, it has been demonstrated that the traditional approach of removing stop words before applying LDA is also ignoring part of the information contained in a document. The analyses in section 5.3 have shown that some terms can be treated like stop words since they appear with similar frequency across documents, while other typical stop words can, in some cases, be topic-related. They also produced indications of naturally occurring non-topics in figure 5.2.

Starting off, section 6.1 will explore the phenomenon of natural non-topics by running LDA on different data subsets. Section 6.2 will then introduce an approach to address ubiquitous terms, the *ubiquitous terms LDA* (uLDA). Section 6.2.1 will include the model proposition as well as the mathematical framework and the sampling process. Possible further expansions of the model are then examined in section 6.2.2.

After presenting the new topic models, section 6.3 will introduce a novel experiment that aims to move model evaluation closer to natural language by introducing a human in the loop element to topic generation. This experiment will later be used to evaluate a range of competing models. Finally, the models that will compete in the following comparison in chapter 7 are described in section 6.4.

6.1 Examining the Natural Non-Topic

The extended descriptive analysis in section 5.3 suggested that under certain conditions, LDA can produce a non-topic, i.e., a topic containing

mostly unspecific terms such as stop words. This section presents a short setup that targets that phenomenon and tries to recreate this effect for other data sets in order to find hints to the underlying conditions.

When looking at the difference between the consumer review data sets, which contained no non-topic, and the *Brexit* data set, which produced a non-topic, the first characteristics standing out are the differences in document length and corpus size. If this drives the occurrence of a natural non-topic, pruning the tent data set to similar dimensions should produce comparable results.

However, simply removing short texts would leave it unclear if the effect is due to document length or corpus size. Therefore, sub-samples were created by selecting similar-sized documents from the tent data set, resulting in 3 data sets of about 200,000 tokens each but varying document size. As a further check for robustness, the analysis will also be repeated on the Reuters data set (UCI 1999), where two subsets were created. Both have a higher average document length than the tent subsets. However, one contains almost three times as many documents as the other. This distinction will help separate the effects of average document length and corpus size. Table 6.1 shows the characteristics of these data sets.

Table 6.1: Number of word tokens per data set.

	Tent Subset			*Reuters*	
	#1	#2	#3	#1	#2
token count	200'018	200'012	200'270	552'071	267'539
document count	3'905	1'221	456	1'148	407
Avg. doc size	51.2	163.8	439.2	480.9	657.3

These data sets are then used to estimate LDA, and the results are examined towards

1. how stop words are distributed across topics,
2. how topics are distributed across documents, and
3. given that there is a candidate for a natural non-topic, how words are distributed within it.

If these corpus dimensions are a contributing factor, the model results for the tent subsets should include a non-topic at some point along the line of increasing restrictions. A similar result is expected for the Reuters data set. The results of this sub-study are displayed in section 7.3.

6.2 Extending Topic Models using the Ubiquitous Terms Concept

This section will provide an extension to standard LDA that incorporates the use of ubiquitous terms. This is done with the objective of utilizing information that was formerly discarded by stop word removal. First, section 6.2.1 will provide the proposition of a new model that includes ubiquitous terms. It is followed by a section discussion further expanding the new model by combining it with other existing approaches (6.2.2).

6.2.1 Latent Dirichlet Allocation with Ubiquitous Terms

The new model that will be proposed by this thesis adapts latent Dirichlet allocation by modeling terms that appear topic-independent and group them into a "non-topic." This way, topics can be de-cluttered from too frequent terms while still keeping the information conveyed by topic-related "stop words." The stochastic implementation also provides a tool to deal with the topic/non-topic ambiguity of specific terms.

The primary assumption of the new model is that a particular set of words exists that is not part of any of the given topics but instead belongs to a "non-topic." Words within this non-topic are equally distributed amongst documents since the document's topic distribution has no impact on their frequency. Mathematically, the non-topic itself is a word distribution vector, like any "normal" topic in LDA. The model's notation is extended from LDA and is also included in the list of symbols on page xv.

Whether or not a specific word w is a topic word or a non-topic word will be denoted by a binary classifier τ, with $\tau = 1$ indicating topic words. Since non-topic words are, by definition, equally frequent across all documents, the probability of a term w being a non-topic word must be constant independently of the respective document, which means that draws of τ follow a Bernoulli distribution with a constant probability.

When the probability for a non-topic term is defined as $1 - \delta$, this leads to a distribution of τ according to

$$\tau \sim \text{Ber}(\delta). \tag{6.1}$$

Prior to model estimation, the frequency of non-topic words in a given corpus is unknown. Additionally, there is no reason to believe that this frequency is constant across all documents of one language. Terms such as "tent" in the *tent* data set have shown that. The different levels of complexity that language can have are also contributing factors. It is therefore advisable to estimate δ for each data set. In a hierarchical Bayes model, this is done by providing a prior distribution. Since $\delta \in (0, 1)$, a natural choice is the Beta distribution

$$\delta \sim \text{Beta}(\gamma), \tag{6.2}$$

with γ being a 2-tuple of (γ_1, γ_2) since the Beta distribution takes two shape parameters.

Given τ, the draw of z is unchanged from LDA if $\tau = 1$. However, if $\tau = 0$, there is no respective z, and w is generated from a non-topic multinomial distribution denoted by ψ:

$$w_{\tau=0} \sim \text{Multinomial}(\psi), \tag{6.3}$$

where ψ itself is Dirichlet distributed with a fixed prior β_ψ, which basically mimics a topic distribution but could potentially have a different prior:

$$\psi \sim \text{Dir}(\beta_\psi) \tag{6.4}$$

Since the uLDA is, like all topic models, a generative model, one clear way of describing it is its data generation process. In analogy to the description of LDA in section 3.1, the process can be described as follows:

Algorithm 2: Data generating process of the ubiquitous terms model

Draw a topic-term probability $\delta \sim \text{Beta}(\gamma)$
Draw a ubiquitous term distribution $\psi \sim \text{Dir}_V(\beta_\psi)$
for *each topic* t **do**
  Draw a word distribution $\phi_t \sim \text{Dir}_V(\beta_\phi)$
for *each document* d **do**
  Draw a vector of topic shares $\theta_d \sim \text{Dir}_T(\alpha)$
  for *each word* $w_{d,n}$ *in document* d **do**
    Draw a topic/non-topic classifier $\tau_{d,n} \sim \text{Ber}(\delta)$
    if $\tau_{d,n} = 1$ **then**
      Draw the topic assignment $z_{d,n} \sim \text{Multi}(\theta_d)$, $z_{d,n} \in \{1, \ldots, T\}$
      Draw a term $w_{n,d} \sim \text{Multi}(\phi_{z_{d,n}})$, $w_{d,n} \in \{1, \ldots, V\}$
    if $\tau_{d,n} = 0$ **then**
      Draw a term $w_{d,n} \sim \text{Multi}(\psi)$, $w_{d,n} \in \{1, \ldots, V\}$

The main difference to latent Dirichlet allocation is that in uLDA, $\tau_{d,n}$ is introduced as an indicator for whether the corresponding word $w_{d,n}$ is a topic-specific word or a term belonging to the "non-topic." LDA, as proposed by Blei, Ng, and Jordan (2003), can be seen as a specific form of uLDA. If $\delta = 1$, the new model is equivalent to LDA, since every word is a topic-specific word and will be drawn from the respective multinomial Distribution Multi($\phi_{z_{d,n}}$). In the case of $\delta = 0$, only "non-topic" terms exist, which are distributed according to ψ. However, the term non-topic becomes meaningless at that point since all words in the corpus are drawn from its probability vector ψ. This case, in turn, is equivalent to LDA with only one topic. It then follows that ψ is the word distribution for all documents since every document consists only of that topic.

As before, the process can be visualized in a directed acyclic graph (DAG). This representation is shown in figure 6.1. Again, the shaded node w is the only observed variable, while the unshaded nodes represent latent variables and α_θ, β_ϕ, β_ψ, and γ_δ are fixed priors.

This generative process can be used to derive the sampling process for model estimation. This thesis uses MCMC with Gibbs sampling to estimate latent variables. As explained in section 3.1, this method is based on sampling each variable depending on all other model variables and then repeating this step until the model is converged.

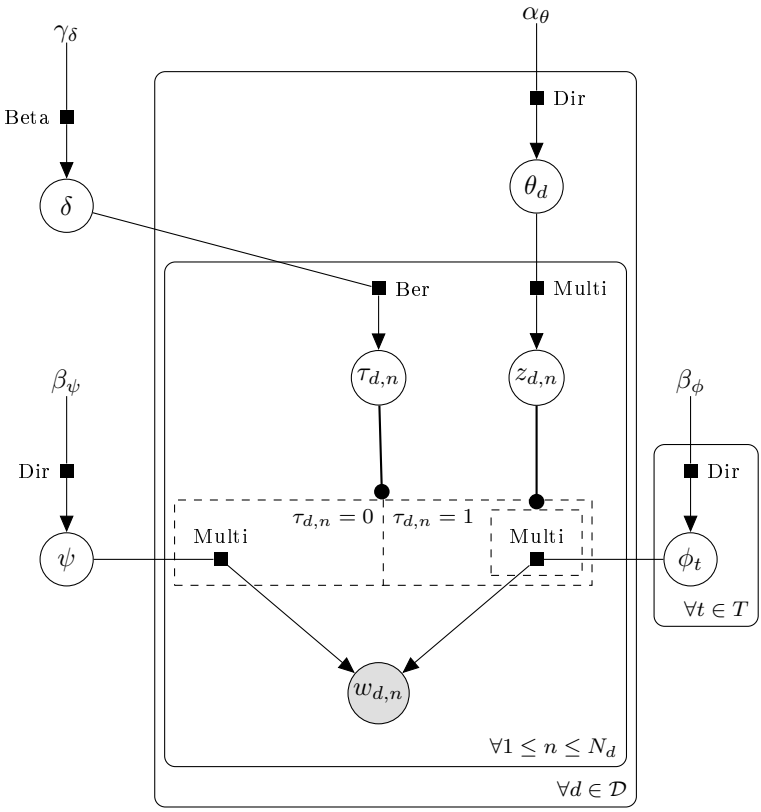


FIGURE 6.1: DAG for the proposed *ubiquitous term LDA*.

In the case of uLDA, the distribution of all parameters can be split up as follows:

$$\begin{aligned} &p\left(w, z, \theta, \phi, \alpha, \beta_\phi, \psi, \beta_\psi, \tau, \delta, \gamma\right) = \\ &= p(w \mid z, \tau, \psi, \phi) \cdot \\ &p(z \mid \theta) \cdot p(\theta \mid \alpha) \cdot p(\tau \mid \delta) \cdot p(\delta \mid \gamma) \cdot p(\phi \mid \beta_\phi) \cdot p(\psi \mid \beta_\psi) \end{aligned} \tag{6.5}$$

Since α, β_ϕ, β_ψ, and γ are fixed priors, equation 6.5 leads to the conclusion that the model estimation requires sampling the following parameters:

τ topic/non-topic indicator on word level

z topic for topic terms on word level

δ topic word probability

ϕ topic word distributions

ψ non-topic word distribution

θ topic distribution on document level

Since the draws are repeated, the order in which the parameters are sampled is irrelevant and could be changed freely.

For τ, the draw depends on the likelihood of w being (any) topic term or a non-topic term. The sampling equation can be derived[1] as:

$$p(\tau_w = 1 \mid \phi, \psi, \delta) \propto \frac{\phi^w \times \theta_d \cdot \delta}{\phi^w \times \theta_d \cdot \delta + \psi^w \cdot (1 - \delta)} \tag{6.6}$$

In this case, ϕ^w denotes the probability of w across all topics in ϕ.

If $\tau_i = 1$ for a specific token w_i, the same formula as in classic LDA can be used to sample topic z_i. In that formula, the draw of z_i depends on z_{-i}, i.e., on all *other* draws of z at that point, and on the token w_i. However, z can be integrated out of the equation, which leads to the draw being dependent on ϕ and θ:

$$p(z_i = t \mid w_i, \phi, \theta) \propto \phi^{w_i} \cdot \theta_d \tag{6.7}$$

[1] The derivations for all sampling equations can be found in Appendix B.

If $\tau_i = 0$, there is no draw of z_i. The distributions θ, ϕ, and ψ are, in analogy to LDA, simple Dirichlet draws with the respective word counts and priors as parameters:

$$p(\theta_d \mid z, \tau, \alpha) \propto \text{Dir}(C_{d,t}^{DT} + \alpha) \tag{6.8}$$

$$p(\phi \mid w, z, \beta_\phi, \tau = 1) \propto \text{Dir}(C_{w,t}^{WT} + \beta_\phi) \tag{6.9}$$

$$p(\psi \mid w, \beta_\psi, \tau = 0) \propto \text{Dir}(C_{w}^{W0} + \beta_\psi) \tag{6.10}$$

The draw of θ is marked as dependent on τ since the value of $\tau = 1$ is necessary for a value of z to exist.

Finally, δ is sampled from a Beta distribution dependent on the current counts of topic and non-topic words and the prior γ:

$$p(\delta \mid \tau, \gamma) \propto \text{Beta}(\sum C_{w,t}^{WT} + \gamma_1, \sum C_{w}^{W0}, +\gamma_2) \tag{6.11}$$

Using these equations, the model is implemented in the statistical programming language R. The respective code can be found in Appendix D.

6.2.2 Expanding on the uLDA Model

Section 6.2.1 has introduced the *ubiquitous terms* extension for classic LDA, which essentially changed the generative process by inserting an additional step before generating w. The simplicity of this approach is also its strength since it enables the inclusion of a non-topic into other extensions of LDA. This section will provide some examples that seem suitable for such an extension and demonstrate this using the case of SC-LDA as defined by Büschken and Allenby (2016).

Extending uLDA

The LDA derivative models most suitable for extending with ubiquitous terms are arguably those that leave the draw of z unchanged. If the sampling of z and w is unchanged from LDA, the inclusion of τ in the process can be copied from section 6.2.1 and inserted before the topic draw. Since it depends only on θ, ϕ, ψ,δ, and w, it is unchanged by any model alteration that happens before the draw of the topic distribution θ. Examples for such models are the latent co-clustering by Shafiei and Muios (2006), the keyword dependant topic distributions as proposed by

Ramage, Dumais, and Liebling (2010), and the author-topic model by Rosen-Zvi et al. (2010). These models alter the draw of θ by introducing some hidden associations between documents. However, since everything happening after that is identical to LDA, the implementation would be relatively simple.

For models that alter the draw of z, the implementation could prove significantly more difficult. However, there are some interesting LDA extensions that could benefit from incorporating ubiquitous terms. For example, the model by Wallach (2006) introduces a dependency of z on the previous term. Implementing a non-topic could enable the model to profit from conjunctive adverbs that might be non-topic related but could indicate a topic change. Gruber, Rosen-Zvi, and Weiss (2007) model the topic sampling to only occur at specific points, while all other values for z are set to match the most recent draw. In this case, introducing a non-topic might enable the model to skip over ambiguous terms that might impact the likelihood of z being the topic of a "topic chunk." However, in both cases, it is impossible to talk about implications without actually doing the math to derive the new sampling procedure. Since those extensions are not the main focus of this thesis, the problem is left to another study to explore.

Lastly, the SC-LDA by Büschken and Allenby (2016) is a suitable candidate since it samples topics on a sentence level, which makes it somewhat similar to the model of Gruber, Rosen-Zvi, and Weiss (2007). When taking a closer look, it turns out to be a particularly interesting case. The sentence constraint leads to *chunks* of text being assigned one topic. This normally would require removing possibly ambiguous terms such as stop words since those appear in most sentences independent of the underlying topic. Otherwise, terms such as "the" would appear at the top of nearly all topic word lists.

Nonetheless, using the *ubiquitous terms* extension, the model can split a sentence into two subsets of terms. One subset, the topic terms, will be used to determine that sentence's topic. The second subset consisting of non-topic terms can be sampled independently and thus do not influence the topic draw. The following paragraphs will provide details on the implementation of this extension.

SC-uLDA Model Proposition

This section will be concerned with implementing ubiquitous terms into the sentence constraint LDA model by Büschken and Allenby (2016). Again, the notation can be found on page xv. In analogy to their model and uLDA, the generative process can be defined in Algorithm 3.

Algorithm 3: Data generating process of sentence-constrained uLDA

Draw a topic-term probability $\delta \sim \text{Beta}(\gamma)$
Draw a ubiquitous term distribution $\psi \sim \text{Dir}_V(\beta_\psi)$
for *each topic* t **do**
  Draw a word distribution $\phi_t \sim \text{Dir}_V(\beta_\phi)$
for *each document* d **do**
  Draw a vector of topic shares $\theta_d \sim \text{Dir}_T(\alpha)$
  for *each sentence* s **do**
    Draw the topic assignment $z_{d,s} \sim \text{Multi}(\theta_d)$, $z_{d,s} \in \{1, \ldots, T\}$
    for *each word* m *in sentence* s **do**
      Draw a topic/non-topic classifier $\tau_{d,s,m} \sim \text{Ber}(\delta)$
      if $\tau = 1$ **then**
        Assign topic $z_{d,s,m} = z_{d,s}$
        Draw a term $w_{d,s,m} \sim \text{Multi}(\phi_{z_{d,s}})$, $w_{d,s,m} \in \{1, \ldots, V\}$
      if $\tau = 0$ **then**
        Assign no topic
        Draw a term $w_{d,s,m} \sim \text{Multi}(\psi)$, $w_{d,s,m} \in \{1, \ldots, V\}$

When comparing this process with the uLDA process, it is clear that the changes are minuscule. The main difference is the shift of the topic draw from word level to sentence level. The directed acyclic graph in figure 6.2 also visualizes this adjustment of the process.

Although the difference in the generative process is minor, the adaptation does get more complicated when turning to the actual sampling process with Gibbs. The difference in sampling lies in the draws of τ and z. The topic term indicator τ is sampled on the word level. Other

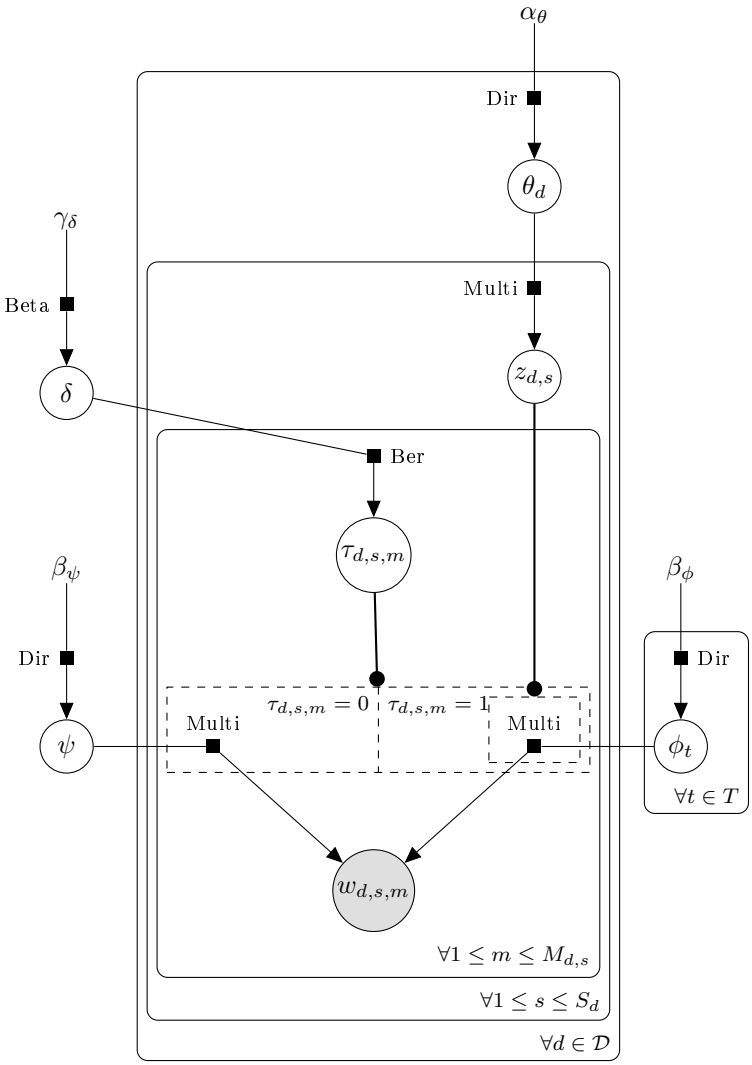


FIGURE 6.2: DAG for ubiquitous term extension to the SC-LDA. The main difference to *uLDA* is the introduction of the sentence plate, which now contains the sentence level draw of z.

than in uLDA, this time, the sampling is conditioned on the underlying sentence topic $z_{d,s}$, which leads to the equation

$$p(\tau_{d,s,m} = 1 \mid z_{d,s}, \phi, \psi, \delta) = \frac{\phi_{w_{d,s,m},z_{d,s}} \cdot \delta}{\phi_{w_{d,s,m},z_{d,s}} \cdot \delta + \psi_w \cdot (1 - \delta)}. \tag{6.12}$$

Based on these values, z is sampled for each sentence with the combined likelihood $\phi_w \times \theta_d$ of all topic words ($\tau_w = 1$) within that sentence. If a sentence contains only non-topic terms, z is sampled from the prior θ following

$$p(z_{d,s} = t \mid \mathbf{w}_{d,s}, \tau_{d,s}, \theta, \phi) = \prod_{m=1}^{M_{d,s}} \left(\phi_{t,w_{d,s,m}}\right)^{\tau_{d,s,m}} \cdot \theta_{d,t}. \tag{6.13}$$

Note that the effect of τ in the exponent is to shrink all values of ϕ for which $\tau_{d,s,m} = 0$ to the neutral element 1. The draws of θ, ϕ, ψ, and δ are unchanged from uLDA (see equations 6.8 – 6.11) and will not be repeated at this point. Overall, this (from an implementation standpoint) simple extension to uLDA can provide an additional benefit by imposing grammatical structure on the data. As such, it will also be evaluated in chapter 7 and discussed in chapter 8.

6.3 Experiment: Approaching Natural Language Topics

The experimental part of this thesis will be concerned with the evaluation of topic model results. When trying to move topic models closer towards natural language, there is reason to question whether or not established evaluation methods are still suitable or if there is a need for a more natural-language-based evaluation. This section will present an experiment that can serve as one first step in that direction.

When evaluating topic models, there are various possible methods that all have their respective costs and benefits. The different approaches can be grouped into three categories: computer-based scoring measures, human-based scoring measures, and application task performance. While computer-based scoring is the cheapest, since it only requires computational cost, it also is considered the least precise. Most

papers that try to construct scoring measures try to quantify topic quality based on the topic's most prevalent words, but the new measures are then usually pitched against human-based scoring as a gold standard to evaluate them. Human-based scoring is usually done by presenting the most prevalent terms of a topic to a human worker to directly or indirectly obtain a rating. As mentioned, this is considered the best method for rating topic quality, but it is also resource-intensive, costing both time and money. The third method is to do away with the topic quality and focus on a task that the model is supposed to perform, for example, document retrieval. However, that would completely ignore the composition of topics.

All three categories have in common that they apply after the model has produced its output. They take the resulting topics and topic distributions and use them to generate a value quantifying their usefulness. None of the approaches tries to evaluate how well the model actually represents the documents of the corpus and the topics within them. This is not surprising since there is no "ground truth" available to compute a fit measure. Although the model implies that the author had specific topics in mind when writing each word, that information is lost to the researcher. However, following the idea behind human-based scoring, which is that human perception is the closest thing to "ground truth," such information could, in theory, be reconstructed. This task is the focus of the following experiment.

6.3.1 Experimental Design

The main idea behind the experiment is arguably trivial. It borrows the topic model assumption that a document is comprised of a range of different topics, and each part of the document can be allocated to one of those topics. This means that each word within the corpus can be assigned a specific topic. If this is done for all the words in all the documents, the result is a vector of topic assignments of the same size as the corpus itself. This is equal to the $\mathbf{z}$ vector generated by topic models. The question now is: what if these word-topic-assignments were in fact conducted by a human. Under the premise that human understanding is "ground truth" for the output generated by an automated model, this would provide a reference point of "natural topics" that can be used to compare with model output. These topics would then be based on natural language that is not limited by pre-processing.

The experiment is approached by taking the above assumption and creating rules and guidelines for a human worker to follow. The following enumeration represents the protocol provided to participants.

Human in the Loop - Protocol

Goal:

1. Generate a set of N topics of which all documents are comprised.
2. Generate a unique(!) assignment of each word of a document to one of these N topics.

Restrictions:

3. The number of topics should be fixed
4. Every word can only be assigned to one topic

Clarifications

5. Concerning (3), there should be some consideration about the topics ahead of the labeling (e.g., after reading about 20 documents). The number of topics should not be changed "on the fly". Instead, topics should be merged or divided if necessary.
6. All participants should agree on a shared value of N to facilitate comparability. This agreement does not extend to topic content.
7. Topics will probably appear in chains of words. This is not mandatory. Sentences can be subdivided to assign several topics. Item (4) always holds true.
8. Topics should not be too specific. For example, the topic "bathroom" is preferred over the collection of topics "bathtub", "shower", "toilet", "sink", and "towels".
9. Topics should be related to content and not to grammar. For example, "quality of the product" could be a suitable topic, while topics such as "double negation" or "conditional statement" are less helpful.

10. In accord with (8) and (9), sentences with different meanings but similar wording could still be assigned to one topic. For example, the sentences "The bathroom was very clean." and "The bathroom was not very clean." could be both assigned to the topic "bathroom". It is not imperative to define the two topics "clean" and "not clean".

As it can be seen here, this experiment's objective is to get the unique assignment for each term within the corpus to one of the T topics. It should also be noted that this means that it is not the goal to provide some form of summary, classification, or ranking. Although this procedure will produce topic labels, these are not used for further analysis. These labels are merely necessary for enabling the worker to identify their own topics when working through the text.

The restrictions are intended to guide the user to a result comparable to topic models. The fixed number of topics is intended to avoid an inflation of topics and mimic topic model behavior more closely. Since the number is still determined by the user, there is enough flexibility to allow for enough topics to represent the corpus. The second restriction is necessary to reinforce unique assignments. Especially with a limited number of topics, preliminary studies have shown that there still can be ambiguous parts within documents where users might be tempted to assign more than one topic.

The clarifications were introduced to give participants a better feel for the desired outcome and further streamline the process. Point (5) aims to the number of topics being determined before labeling. Ideally, the user would read the whole corpus before deciding on the optimal number of topics. However, this was determined infeasible due to the additional workload. Instead of adding topics whenever a new issue occurs to the reader, participants are encouraged to redefine existing ones to include new information.

Clarification (6) was defined to facilitate the comparison of the results across different participants. It can facilitate further analysis when looking at variance between different results. Although it could be argued that this is not a vital restriction, it was included for comfort.

Points (7) to (10) are concerned with topic formation. The first of them is aimed to raise awareness that this experiment's goal is obtaining labels on a word level. Preliminary studies have shown a tendency to orient on sentence structures. Although topics might be chains of words, participants are encouraged to break these chains into smaller parts if

it better fits a topic. This rule can also help to avoid ambiguity. The remaining points try to clarify what can be considered a topic. Point (8) is intended to motivate smaller topic numbers, while (9) was introduced as a clarification after some testing revealed that participants might not necessarily be familiar with the meaning of "topic" in a topic model context. Finally, (10) is also aimed to help find a reasonable scope for topics.

These instructions were given to study participants to follow, together with the unedited three corpora introduced above, namely the *tent*, *dogfood*, and *Brexit* data sets. Following the protocol, three sets of topic labels were produced and will be examined in the results chapter (7). Implications of these results, as well as learnings from this experiment, are elaborated in chapter 8.

6.3.2 Label Utilization

Since this experiment is a novel approach, it should be clarified what is to be gained by the resulting topics. The first use will be a descriptive analysis to gain some understanding of how the labeling was performed. This exploration includes typical metrics such as the average size of topics and also more specific phenomena such as the points where topics are changed. The second use of the labels will be comparing these topics to those resulting from actual topic models. This part is a bit more complex and will therefore be explained in detail.

When comparing the human labels to topic model results, a few steps are necessary beforehand. The main issue here is that topic models are agnostic towards topic labels and topic order. That means the model does not contain information on which topic is concerned with which content except for the word probability vector. Also, the stochastic characteristics of MCMC and the randomization of starting values can lead to different sorting each time the model is run. For example, when the human labels provide a topic titled "tent size," there is no direct information given by the model whether or not its output contains a topic that fits this title, and if there was, which of its topics would correspond to this. Moreover, the position of said topic could change after every model run.

These issues lead to some preparation having to be done to match the human labels with topic model results. Therefore, the objective could be:

> For each Topic z^H produced by human labeling, find a matching topic z^T from the topic model output.

However, this is not always possible. No mechanism is in place that guarantees that a matching topic exists. Therefore, the best approach is to find the *closest* topic, with the definition of "closest" put aside for the moment. However, the next arising problem is that assigning each topic to its closest counterpart does not necessarily produce a bijective mapping. Since no topics should be discarded, assigning some topics to sub-optimal counterparts might be necessary. This compromise leads to the new formulation of the objective as:

> Find a bijective mapping
>
> $$f\colon z^H \mapsto z^T, \qquad z^H, z^T \in T$$
>
> that maximizes the overall similarity between both topic sets.

The following pages will explain how this task was approached within this study.

The first step is to produce a topic representation of the human coding comparable with the topic model output. With a given z, the count matrices C^{WT} and C^{DT} can be calculated by counting the occurrences of word-topic combinations or topic-document combinations, respectively. Given those matrices, implicit estimates for ϕ and θ can be obtained via:

$$\phi^H_{w,z} = \frac{C^{WT}_{w,z}}{\sum_{i=1}^{W} C^{WT}_{i,z}} \tag{6.14}$$

$$\theta^H_{d,z} = \frac{C^{DT}_{d,z}}{\sum_{i=1}^{T} C^{DT}_{d,i}} \tag{6.15}$$

These approximations facilitate a direct comparison to topic model results. However, there is still one obstacle left. There is no unambiguous mapping of model-based topics T^M and human-based topics T^H. A solution for this problem would be a method that can match similar topics and thereby generate a mapping that can be used to compare the results. In the literature, there are different approaches to comparing topics. Greene, O'Callaghan, and Cunningham (2014) propose comparing two top-word lists by *Average Jaccard*, which is a modification of the Jaccard score that puts more weight on the higher-ranked terms. M.

Steyvers and T. Griffiths (2010) use the symmetrized Kullback Leibler divergence. A similar alternative is the Jensen-Shannon distance mentioned by Heinrich (2009). After a short exploratory analysis, the latter measure was selected for this experiment. The distance between two distributions X and Y is defined as:

$$D_{JS}(X||Y) = \frac{1}{2}\left[D_{KL}(X||M) + D_{KL}(Y||M)\right] \tag{6.16}$$

where $M = \frac{1}{2}(X+Y)$ and $D_{KL}(X||Y)$ is the Kullback-Leibler divergence between X and Y as produced by

$$D_{KL}(X||Y) = \sum_{n=1}^{N} p(X=n)\left[\log_2 p(X=n) - \log_2 p(Y=n)\right] \tag{6.17}$$

With this step solved, another problem arises concerning the uLDA and SC-uLDA models. Since some of the words are pulled out by the non-topic, ubiquitous terms are ranked lower in ϕ^M than they would be otherwise, while the human coding does not perform any separation of that sort. Thus, the high-ranking ubiquitous terms in ϕ^H could be distorting the matching process. This issue can be countered by re-integrating ubiquitous terms into the topics before matching via

$$\phi_i^{M*} = \delta \cdot \phi_i^M + (1-\delta) \cdot \psi^M, \tag{6.18}$$

which improves the matching results across all data sets and (ψ-related) models. Note that, from a statistical standpoint, this is not necessarily a valid integration of the two probabilities. Nevertheless, the only point of this equation is to better match the model topics ϕ^M, which had stop words extracted, with the human coded topics ϕ^H, for which no stop word removal has taken place. This equation is not suggesting that ϕ_i^{M*} is a true representation of word probabilities.

After a topic model is fitted with the number of topics equal to T^H, this measure is calculated for all possible topic matches. Since the matching of topics can be viewed as an assignment problem, which has been well-known for a long time in logistics, there are various approaches to solve it. In this case, the resulting matrix is used to optimize the topic mapping via the Hungarian method (Kuhn 1955; Munkres 1957), which is a widely used, simple, and fast algorithm to solve the assignment problem.

When each topic z_i^M has been assigned to a topic z_j^H, the model hit rate can be calculated. This is done by simply averaging the number of hits. Unfortunately, this method encounters a problem when trying to process *ubiquitous terms* models such as uLDA. The output of those models does not contain a value of z for every word since non-topic terms have no assigned topic. Just including the non-topic as another topic is no sensible option since the terms are explicitly different from topic terms. This leaves three options to treat non-topic tokens. The first option would be counting them as a "miss." This choice would heavily diminish the resulting hit rate and penalize the model for working as intended. For the second alternative, the non-topic tokens could be counted as a hit. However, this would arguably inflate the hit rate. Therefore, the third option was chosen, in which non-topic tokens are removed altogether before calculating hit rates. Thus, only the share of actual topic word hits is taken into account. With the number of tokens that have an assigned topic z denoted as N^T, the equation can be written as:

$$N^T = \sum_{i=1}^{N} \tau_i \tag{6.19}$$

$$\mathbf{1}_z(z, z^*) = \begin{cases} 1, \text{if } z = z^* \\ 0, \text{otherwise} \end{cases} \tag{6.20}$$

$$\text{hitrate}(M, H) = \frac{1}{N^T} \sum_{i=1}^{N^T} \mathbf{1}_z(z_i^H, z_i^T) \tag{6.21}$$

The baseline for this value is an entirely random assignment of topics, which would lead to a hit rate of $\frac{1}{T}$, e.g., 5% for $T = 20$.

With the objective to reduce noise, the values z_i^M for all tokens w_i in the corpus are not simply fetched from the most recent Gibbs sampling step. They are instead generated from the posterior distribution of z, which is estimated by looking at the S most recent draws of z for each token. First, each topic gets assigned its share of all S draws by counting the occurrences of z and dividing the value by S. The topic with the highest share is chosen as z^M, with ties broken randomly.

Algorithm 4: Generating the smoothed topic label results z^M

Generate a vector v of length $T + 1$ with all values set to 0 and indices ranging from 0 to T.
for *each word* $w \in$ *corpus* $\mathcal{C}$ **do**
  for *each draw* $s \in 1, \ldots, S$ **do**
    if $z_{w,s}$ *does not exist* **then**
      set $z_{w,s} = 0$
    $v[z_{w,s}] = v[z_{w,s}] + 1$
  $z_w^M = \text{argmax}(v)$

The results from this procedure are hit rates for every model on each data set. Those will be presented in section 7.3.

6.4 Models Selected for Comparison

The main focus of this thesis is put on the uLDA extension. However, when trying to quantify the performance of the new approach, reference models are needed. Since there is plenty of research on the performance of existing topic models, the aim is a direct comparison between the existing models and the extension. Within the scope of this thesis, that leads to a total of four models:

LDA The standard LDA model, as defined in section 3.1 (see Blei, Ng, and Jordan 2003).

SC-LDA The sentence constraint model, as defined by Büschken and Allenby (2016).

uLDA The ubiquitous terms LDA, as it was introduced in section 6.2.1.

SC-uLDA The ubiquitous terms extension to the SC-LDA, as introduced in 6.2.2.

The actual implementation of all these models with R can be found in Appendix D.

Chapter 7

Results

This chapter contains the results for the different analyses outlined in chapter 6. First, the results for the analysis of possible natural non-topics are presented in section 7.1. Afterward, section 7.2 includes the model results from all four models that were selected earlier (6.4). Finally, the results for the experiment to validate topic model results using a human in the loop are presented in section 7.3.

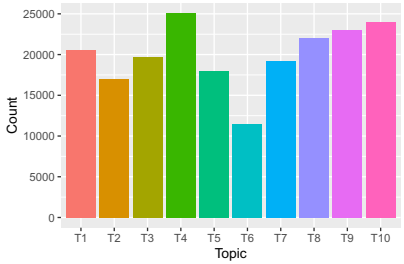
7.1 Naturally Occurring Non-Topics

The analysis presented in section 6.1 produces interesting results. The emergence of a "natural non-topic" could successfully be reproduced for data sets containing longer documents.

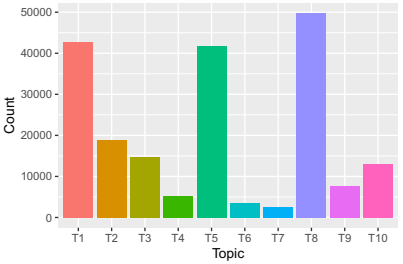
Figures 7.1 show the distribution of topics over the corpus for all data sets. It is clear that those corpora with longer documents contain a dominant topic. For this to be a "natural non-topic," the following has to hold true:

1. It appears in high frequency across all documents.
2. It has only little variance in frequency across documents.
3. It contains mainly typical function words or non-informative terms.

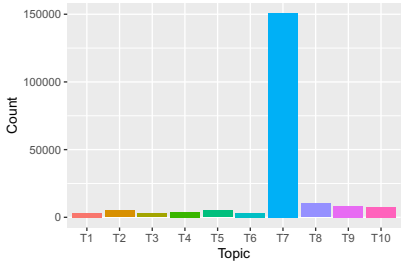
Points 1 and 2 can be verified by inspecting Figure 7.2, which plots the respective θ values for each document. The respective topic is prevalent across all texts within the respective data sets. However, this effect is more clearly in the tent data set and less so in the Reuters data set. Figure 7.3 shows the respective word clouds, which confirms point 3. Implications of these results will be discussed in section 8.1



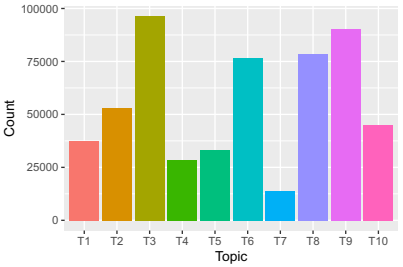
(A) Topic shares for the *tent* data set with an average length of 51.2.



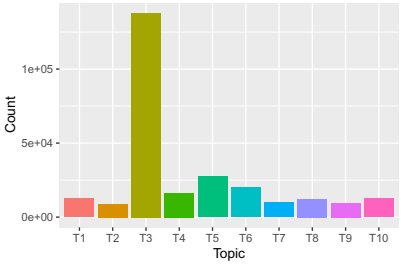
(B) Topic shares for the *tent* data set with an average length of 163.8.



(C) Topic shares for the *tent* data set with an average length of 439.2.

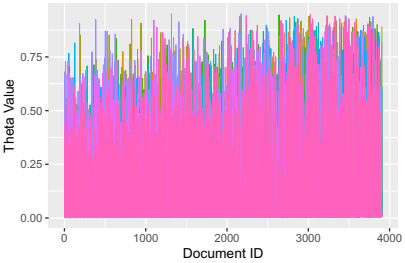


(D) Topic shares for the *Reuters* data set with an average length of 420.6.

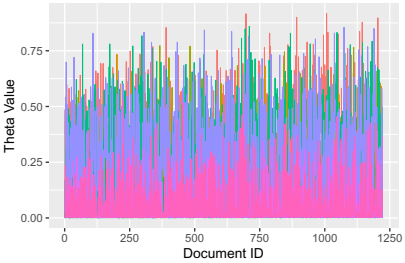


(E) Topic shares for the *Reuters* data set with an average length of 657.3.

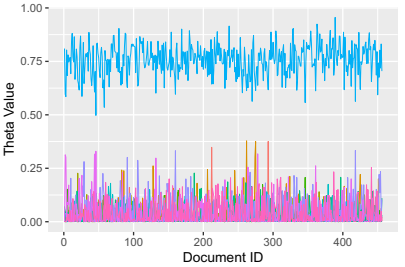
FIGURE 7.1: Shares of topics in each data set.



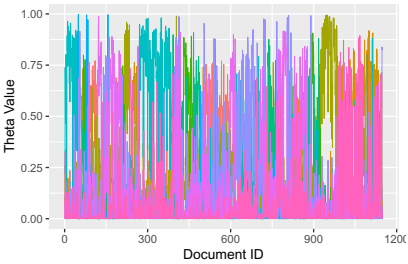
(A) Distribution of theta for the *tent* data set with an average length of 51.2.



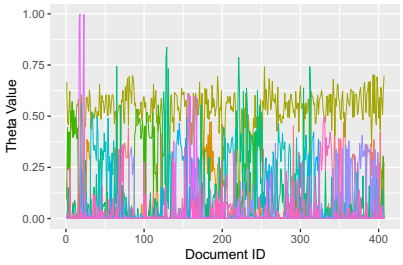
(B) Distribution of theta for the *tent* data set with an average length of 163.8.



(C) Distribution of theta for the *tent* data set with an average length of 439.2.

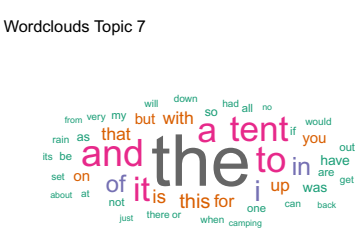


(D) Distribution of theta for the *Reuters* data set with an average length of 420.6.



(E) Distribution of theta for the *Reuters* data set with an average length of 657.3.

FIGURE 7.2: Distribution of theta across all documents within each data set.



(A) Word cloud of the non-topic for the *tent* data set with average length of 439.2.

(B) Word cloud of the non-topic for the *Reuters* data set with average length of 657.3.

FIGURE 7.3: Word cloud of top words from the naturally occurring non-topic.

7.2 Topic Model Results

This section presents a summary of the results of all topic models that were run for this study. After section 7.2.1 introduces the chosen model parameters. Section 7.2.2 presents convergence and model fit, followed by a description of the model's output with both the resulting topics and their distribution.

7.2.1 Model Parameters

The model proposition in section 6.2.1 showed that the model's hyperpriors have to be fixed a priori. For this thesis, so-called flat priors were selected, which means that the impact of the prior on the result is kept minimal. The following values were determined for all models:

$\alpha = 0.1$ as the Dirichlet prior for θ.

$\beta_\phi = 0.1$ as the Dirichlet prior for ϕ.

$\beta_\psi = 0.1$ as the Dirichlet prior for ψ, if applicable.

$\gamma = (3, 3)$ as the Beta prior for δ.

All these parameters are very small compared to the influence of the data itself. The draw of θ is influenced by $+1$ for each word in a document,

which means that the Dirichlet parameter for $T = 5$ and $N_d = 100$ could be, for example

$$\theta_d \sim \text{Dir}(C_{d,*}^{DT} + \alpha) = \text{Dir}(30.1, 40.1, 20.1, 10.1, 0.1).$$

The last parameter that has to be determined is the number of topics T. Unfortunately, estimating the optimal value for T is a difficult task that is already subject to many publications (Greene, O'Callaghan, and Cunningham 2014; Arun et al. 2010; Zhao et al. 2015, to name just a few). Instead of a lengthy digression on different techniques and heuristics, the following paragraphs will discuss the question of whether or not the optimization of T matters in the scope of this work. First, the problem is not an easy one. There have been plenty of different approaches on how to optimize T. For example, Greene, O'Callaghan, and Cunningham (2014) suggest estimating the topic model on corpus subsamples and calculate an "agreement score" between the resulting topics. Arun et al. (2010), on the other hand, turn to single value decomposition of the distribution matrices Φ and Θ. Finally, Zhao et al. (2015) propose a measure based on the change in perplexity when the number of topics is incremented. All these methods have in common that they require estimating the model across a wide range of values for T and comparing the results ex-post. This would make it necessary to estimate M models, with M typically in the range of $50 - 100$, but only end up using one result, which is highly inefficient. In summary, it could be argued that there is no consensus on the "best" approach to determine T, and most techniques are resource-intensive.

Additionally, the results from the human-based experiment in chapter 6.3 provided a value for T that is considered optimal in that context. Since the context of this thesis is not optimizing T, it can be argued that any reasonable number of topics could be fixed for this analysis. So the obvious choice would be to take the values of T that were parts of the experiment results, which are displayed in table 7.1. However, there is a slight theoretical chance that the following results could be an anomaly that appears only at these values of T. This possibility is acknowledged by including short versions of the following evaluations for different values of T in Appendix C, which show that the results are robust towards varying the number of topics.

TABLE 7.1: Number of topics per data set.

Note: These values were results of the human-based experiment as described in section 6.3.

	Data Set:		
	tent	dogfood	Brexit
Number of Topics	28	24	10

7.2.2 Model Output and Fit

Since the parameters were estimated using MCMC, model convergence is an essential issue for evaluating the output. The models were run for 300-2000 iterations, depending on the model, and converted after 150-1500 iterations on average. Table 7.2 shows a breakdown of these values for the different models. Every model uses only the last 100 iterations for inference.

TABLE 7.2: MCMC convergence characteristics for different models.

	Model:			
	LDA	SC-LDA	uLDA	SC-uLDA
Iterations run	1'500	300	1'500	2'000
Conv. after iteration	1'000	150	1'000	1'500
Iterations used	100	100	100	100

Model fit was measured by out-of-sample log-likelihood $\mathcal{L}_{oos}$. For this, a 10% holdout was separated from the training data and used for evaluation. A condition of this sampling step was that the training set had to include the human-coded documents, thus creating a small, neglectable bias. Since holdout data is missing information on θ, z, and, if applicable, τ, calculating the log-likelihood is not trivial. Wallach, Murray, et al. (2009) suggest a variety of possible solutions for this problem. Most approaches have in common that they substitute the missing θ for its prior α, which results in a symmetric distribution when applied in

this study. The implementation chosen for this thesis is the harmonic mean approach, mainly for its manageable computational cost. This method estimates the probability of the data given model parameters, for instance, $P(\mathbf{w} \mid \phi, \alpha)$ for LDA, by generating S samples for that probability and then calculating the harmonic mean according to

$$P(\mathbf{w} \mid \phi, \alpha) \approx \frac{S}{\sum_{s=1}^{S} \frac{1}{P(\mathbf{w}|z^{(s)}, \phi)}} \tag{7.1}$$

for standard LDA models and

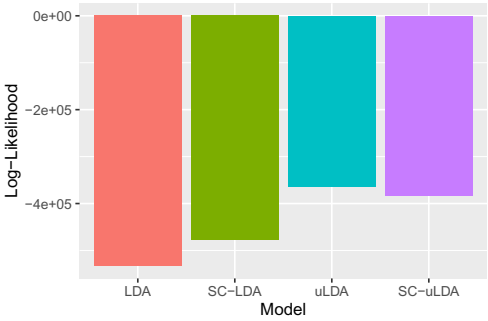
$$P(\mathbf{w} \mid \phi, \psi, \delta, \alpha) \approx \frac{S}{\sum_{s=1}^{S} \frac{1}{P(\mathbf{w}|z^{(s)}, \tau^{(s)}, \phi, \psi)}} \tag{7.2}$$

for ubiquitous term models. The values for $z^{(s)}, \tau^{(s)}$ in turn are generated by Gibbs sampling while holding model parameters such as ϕ, ψ constant. The authors suggest prepending a short burn-in period to the S Gibbs sampling steps. In this paper, the log-likelihood was estimated using $S = 100$ after a burn-in period of 10 iterations. Figure 7.4 shows the resulting $\mathcal{L}(\mathbf{w}) = \log(P(\mathbf{w} \mid \cdot))$ values for the different models on the different data sets.

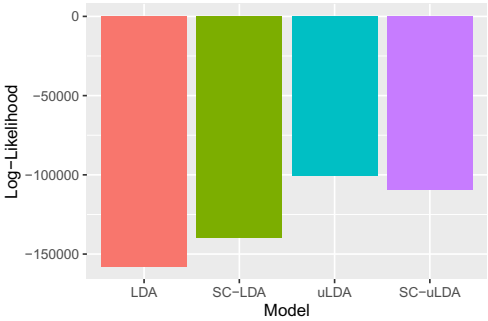
The graphs indicate an advantage for the ubiquitous term models across all data sets, which can be taken as an indicator that the new models generate a better fit on the data. However, the argument could be made that comparing uLDA to LDA on the same data is inappropriate since the latter is not designed to handle ubiquitous terms. Therefore, LDA and SC-LDA were estimated again, with typical stop words removed. Since this means that the corpus size $W = dim(\mathbf{w})$ is reduced, a direct comparison of the log-likelihood is no longer suitable. Therefore, the perplexity of all terms $\mathbf{w}$ was calculated via

$$\text{perplexity}(\mathbf{w}) = \exp\left(-\frac{\mathcal{L}(\mathbf{w})}{dim(\mathbf{w})}\right), \tag{7.3}$$

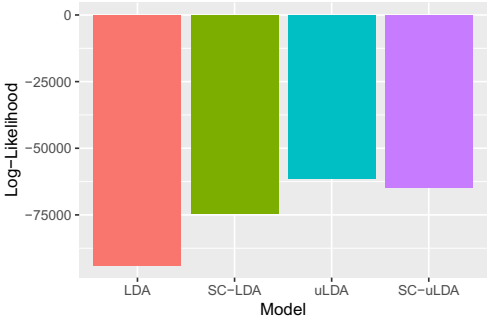
which provides a measure that is relative to the number of tokens $dim(\mathbf{w})$ in the holdout data set. The results of this in figure 7.5 show that ubiquitous term variants still outperform their more standard counterparts, and the fit measures are also higher for standard LDA models on the less preprocessed data.



(A) Out-of-sample fit for the *tent* data set.

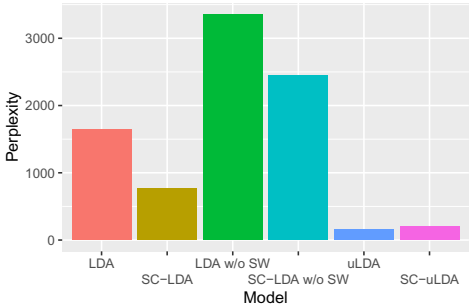


(B) Out-of-sample fit for the *dogfood* data set.

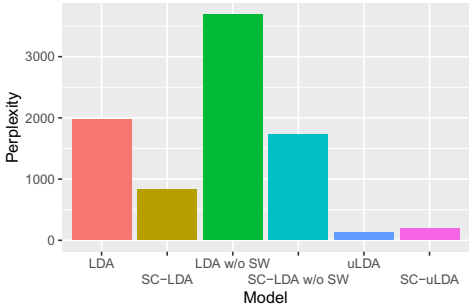


(C) Out-of-sample fit for the *Brexit* data set.

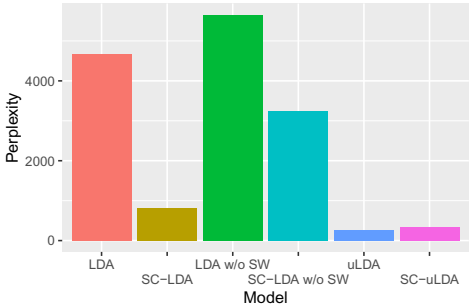
FIGURE 7.4: Comparison of the out-of-sample log-likelihood split by data set and model.



(A) Out-of-sample perplexity for the *tent* data set.

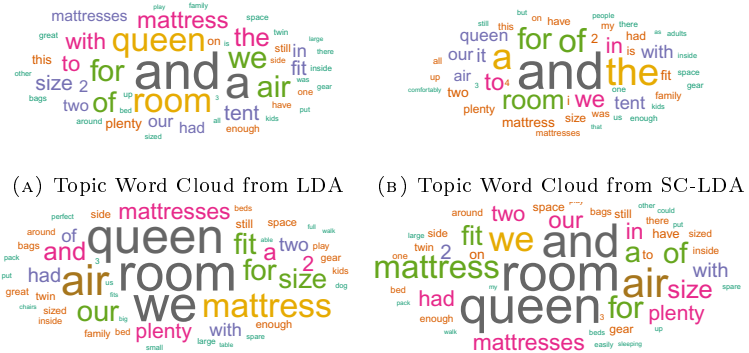


(B) Out-of-sample perplexity for the *dogfood* data set.



(C) Out-of-sample perplexity for the *Brexit* data set.

FIGURE 7.5: Comparison of out-of-sample perplexity by data set and model. Lower values are better.



(A) Topic Word Cloud from LDA (B) Topic Word Cloud from SC-LDA

(C) Topic Word Cloud from uLDA (D) Topic Word Cloud from SC-uLDA

FIGURE 7.6: Word clouds for similar topics of different models in the tent data set. This topic seems to be concerned with the available space for mattresses.

Further exploration of the model results is done by looking at the actual topics that were produced. Figures 7.6, 7.7, and 7.8 show examples of similar topics from all models side by side. Although these are only a few prominent instances, they show that the inclusion of a non-topic tends to give more room to topic-related terms, as they are more prominent in the ubiquitous terms extensions. It is also not surprising that the SC-LDA model has the most cluttered stop words since it is forced to include such terms due to the sentence constraint. Another interesting finding is that the "natural" non-topic mentioned in section 5.3.2, which was an output of LDA when applied on the *Brexit* data set, has a similar effect on the topic terms to the uLDA extension.

The resulting word clouds show that the LDA extensions are working as intended. When comparing the resulting topics, the ubiquitous term variants can reduce the relative importance of stop words ("the," "and") while putting more emphasis on topic-related words ("room," "amazon," "ireland"). Figure 7.8a shows a very similar topic to its uLDA counterpart since the results include a natural non-topic, as will be discussed in 8.1.

In Addition to these established fit measures, the following section will now present the results form the human-based experiment.

(A) Topic Word Cloud from LDA

(B) Topic Word Cloud from SC-LDA

(C) Topic Word Cloud from uLDA

(D) Topic Word Cloud from SC-uLDA

FIGURE 7.7: Word clouds for similar topics of different models in the dogfood data set. This topic seems to be concerned with comparing the price to other retailers such as Amazon.

(A) Topic Word Cloud from LDA

(B) Topic Word Cloud from SC-LDA

(C) Topic Word Cloud from uLDA

(D) Topic Word Cloud from SC-uLDA

FIGURE 7.8: Word clouds for similar topics of different models in the Brexit data set. This topic seems to be concerned with the issues regarding Northern Ireland. Note that LDA has a clearly defined topic due to its naturally occurring non-topic (see section 8.1).

7.3 Human-Based Topics Experiment

This section contains an overview of the results from the human-based experiment presented in section 6.3. The main results of the experiment are the topic labels for each token within each labeled data set. These do not only provide information on how often each term was assigned which topic, but also give insight on the points where topics change. Most of the time, a string of consecutive words is assigned the same topic, implying that the whole sequence is concerned with the same issue. This phenomenon will be denoted as a "topic run." It turns out that topic runs are the most common form of topic labeling in this experiment. Table 7.3 shows some information on how the length of these runs is distributed.

TABLE 7.3: Statistics describing the distribution of "topic run" lengths.

	descriptive measures					
	min.	1st qu.	median	mean	3rd qu.	max,
# token(s)	1.00	7.00	15.00	24.73	31.00	382.00

Since these runs are very common, the points within the text where they end become more interesting. Therefore, these change points are examined closer to find a pattern that could motivate a topic switch. For this, the terms directly preceding and following such a change point are examined, including any punctuation. If punctuation was present directly before the change, it was taken at the most likely reason. This also goes for brackets. If no punctuation was present, the next step was to check for conjunctions immediately before or after the break, which were then taken as the most likely cause. If both terms were conjunctions, the word after the break was chosen. And finally, if none of the previous steps provided a possible reason, the first word of the new topic run was chosen.

The most common reason identified by this procedure is the end of a sentence, represented by either a full stop, a question mark, or an exclamation mark. The second most frequent cause is due to conjunctions, with the most prominent representatives being the words "and," "but," and "as." In third place is the start of a half-sentence, which was chosen

if the previous run was ended by either a comma, a colon, a semicolon, or brackets. Everything else is referred to as "miscellaneous," although it might be interesting that among these terms, the most common ones were clearly related to a specific topic, such as "price," "food," or "ingredients." The percentages are also displayed in table 7.4. Implications of these results will be discussed in 8.3.

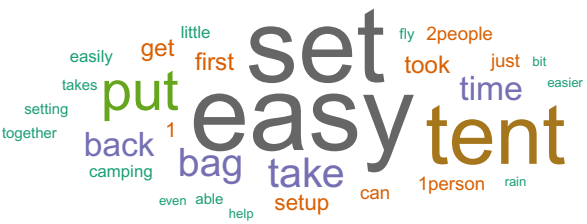
TABLE 7.4: Likely reasons for the change in topic within the labeled data.

	Sentence Boundaries		*Conjunctions*	*Misc.*
	. ! ?	, ; : ()	and, but, ...	price, taste, ...
share of all changepoints	76.7%	8.1%	8.7%	6.3%

Besides topic runs and change points, the labels are also examined concerning how often each topic was chosen. Since each token is assigned exactly one topic, the number of times any token is assigned topic z can easily be counted, which will be referred to as "topic size." It turns out that, especially for the consumer review data sets, the topic sizes vary a lot from topic to topic. Some descriptive measures about the topics can be seen in table 7.5.

The high variance for the consumer review data sets might stem from the relatively large number of small topics. To illustrate this, the topics are sorted from smallest to largest. Then, beginning with the smallest topic, the number of topics that can fit in 1% of the respective corpus size (measured in *number of tokens*) are counted. For the *tent* and the *dogfood* data set, 6 and 4 topics fit in this 1% window, respectively. The *Brexit* data set, on the other hand, does not have a topic as small. The fact that the number of topics is smaller for the latter data set might be a contributing factor.

As described during the experiment design section (6.3.1), all topics were attached with topic labels describing the context. Figure 7.9 shows the description with a word cloud of the respective largest topic, i.e., the topic with the most tokens assigned to it. Since stop words are removed in these word clouds for improved readability, the *tent* topic is missing the term "up," which would otherwise be part of the bigram "set up."



(A) Largest topic for *tent*: "use simplicity"

(B) Largest topic for *dogfood*: "dog likes it / good taste"

(C) Largest topic for *Brexit*: "historical/general background"

FIGURE 7.9: Description and top words of the largest human-coded topics of each data set. Note that stop words were removed for the graphic to improve distinguishability.

TABLE 7.5: Descriptive statistics of the human-labeled topics.

	units	tent	dogfood	Brexit
number of topics	#	28	24	10
size of smallest topic	(tokens)	8	31	2'571
size of largest topic	(tokens)	7'285	5'710	20'505
share of largest topic on all terms	%	0.168	0.160	0.177
topics that fit in 1% of terms	#	6	4	0

The hit rates that can be calculated for the different models and data sets range roughly from 12-24%, notably higher than the baseline of random assignment. That baseline is about 3.6-10%, depending on the data set. Table 7.6 shows hit rates for different models and data sets, including the baseline values. In some cases, such as the sentence-constrained uLDA model on the *tent* data set, the baseline is surpassed many times over.

TABLE 7.6: Average topic hit rates with comparison to baseline results.

	LDA	SC-LDA	Ubi-LDA	Ubi-SC-LDA	Baseline
tent	15.7%	20.5%	17.7%	23.5%	3.6%
dogfood	13.9%	18.4%	16.6%	22.8%	4.2%
Brexit	15.8%	23.6%	17.2%	22.4%	10%

As the results show, ubiquitous term models outperform the traditional counterparts in almost all cases. The only exception is the *Brexit* data set, where the standard SC-LDA performs slightly better than its new extension. This might be connected to the different characteristics of that corpus compared to the other two, which consist entirely of consumer reviews. Those documents tend to be shorter and use less complex language. These results will be analyzed in more detail during the discussion in section 8.

The results of Table 7.6 also show that the hit rates vary from data set to data set. This warrants a closer look in order to assess two possible

contributing factors. First, it is analyzed whether the different data characteristics, such as document length or language complexity, could influence the relative performance of models. The second step is to look closer at a possible connection between correct model hits and the topics identified by workers. These factors are not exclusive, so a mixture of both could impact the results. To find indication for one or both factors, a simple regression is performed, and results are compared.

Additionally, the impact of the different coding on the hit rates is examined. For this, a topic-level hit rate is calculated. Figure 7.10 shows those hit rates for different topics in all three data sets. It shows that, at least for high topic numbers, only a few topics have very high hit rates while other topics are barely identified correctly. This asks for further examination of topic characteristics to find any information on what could drive topic identifiability.

Low hit rates for small topics could be rooted in the fact that the topics are not necessarily shared, but both the model and the human coder create their own set of topics, leading to nuanced differences in smaller topics. However, it is clear that certain topics drive the overall model performance in this task.

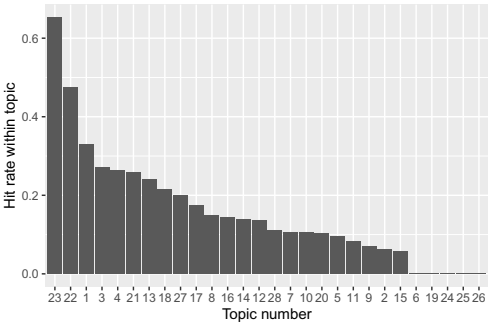
The variance in performance between the different topics begs the question of whether or not identifiable characteristics of these topics exist that drive the hit rates. Answering this could help to formulate what makes a "good" topic. Consequently, the topics were examined for:

- size,
- concentration,
- and similarity to other topics.

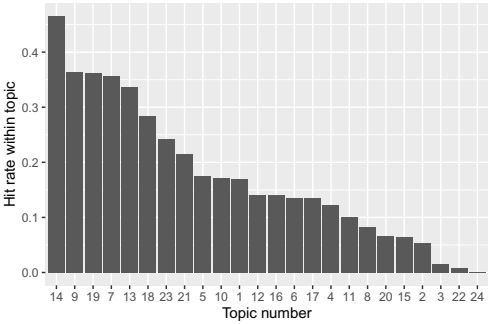
However, there was no significant correlation found between hit rates and any of the above characteristics. The only solid result is that large[1] topics usually have similar, average-sized hit rates while smaller topics seem to be more "hit or miss," meaning they display a higher range in hit rates achieved.

Another possible contributor is variance in the data structure. For this, the hit rates are examined on a document level to identify document characteristics that influence the resulting document hit rate. The chosen exogenous variables are displayed in table 7.7.

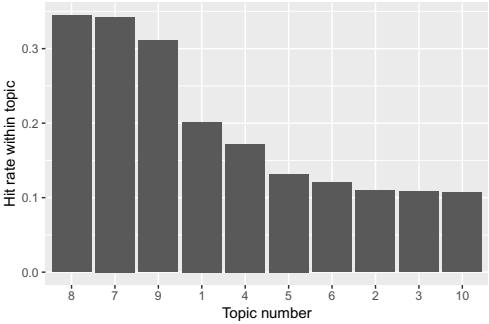
[1] In this context, "large" means that the respective topic was assigned to comparably many tokens within the corpus.



(A) *tent* data set



(B) *dogfood* data set



(C) *Brexit* data set

FIGURE 7.10: Topic-specific uLDA hit rates for the different data sets.

TABLE 7.7: Text characteristics used for the regression analysis of hit rates.

Variable	Explanation
wordcount	The length of a document, measured in number of words.
docvocab	The number of unique words within a document.
wordlength	The average number of characters per word.
sentlength	The average number of sentences per document
share_repeats	The share of words that appear multiple times.
lexical	Lexical classification score.
colemanliau	The Coleman-Liau index, a readability score.

The lexical classification score is used to describe the information density within a text. It is calculated by dividing the number of lexical words by the number of total words in a text. Lexical words are words that have lexical meaning, which is simply determined by matching them against a list of function words. All words not on that function words list are considered lexical words. For this analysis, the R function `lexical_classification` from the package `qdap` was used.

The Coleman-Liau index is a readability score developed by Meri Coleman and T. L. Liau in order to estimate the grade level necessary to comprehend a text (see R. P. Reck and R. A. Reck 2007). It uses the average character length of words as well as the inverse average word length of sentences. With some weighing factors, the total formula is[2]:

$$\mathrm{CL} = 5.88\left(100 \cdot \frac{characters}{words}\right) - 0.296\left(100 \cdot \frac{sentences}{words}\right) - 15.8 \quad (7.4)$$

It should be mentioned that there are plenty of different readability scores used in linguistics, for example, the Flesch Score, SMOG, Lix, or the Gunning Fog Index, to name just a few. However, most of them are very similar and rely in one form or another on sentence length and word

[2]Note that R. P. Reck and R. A. Reck (2007) present a different definition of the Coleman-Liau-Index that seems to correlate almost perfectly with the average number of characters per word. Instead, the original index as proposed by Coleman and Liau (1975) was used here.

complexity (mainly measured by word length). Due to this similarity between these test scores, it would be ill-advised to include several of them in a regression model since they would be heavily correlated. The Coleman-Liau index was chosen due to its relatively simple equation, since it does measure word length based on characters and not based on syllables, and due to its significant influence based on preliminary test models. Although these tests are usually used for longer texts, they can still be applied here since the goal is not actually measuring the readability rather than quantifying text complexity in order to look for a connection to model performance.

Tables 7.8 to 7.10 show the regression results for the different data sets. these outputs will be discussed further in section 8.3.

Besides text complexity characteristics, another possible influence on hit rates that should be examined are the specific prevalent topics within a given document. According to Figure 7.10, the topic shares should significantly influence the score for the high-accuracy topics. This effect should be more pronounced in the *tent* and *dogfood* data sets since the hit rate is less balanced across topics. Note that due to redundancy, the regression models contain one fewer topic than the actual topic model since the shares always add up to 1.

The results in tables 7.11 to 7.13 show that the prevalence of specific topics within a document is highly correlated with the resulting hit rate, while most others do not seem to impact the hit rate at all. This means that, according to the regression models, documents that have a high share of specific topics tend to produce better hit rates than others, which confirms the implications of figure 7.10 that some topics produce higher hit rates than others. As a logical consequence, documents that consist mainly of topics with high hit rates perform better overall. The output tables also provide information on how much of the hit rate variance can be explained with the respective topic shares. The adjusted R^2 ranges from 8% for SC-uLDA on the *Brexit* data to 61% for uLDA on the *tent* data set. This high dispersion of values does not allow for a general interpretation of the explanatory power of topic shares. The phenomenon seems to be connected to either corpus structure or number of topics, or even both.

TABLE 7.8: Regression output for *tent* data set.

	Dependent variable:			
	LDA	SC-LDA	uLDA	SC-uLDA
Constant	0.257^{***}	0.181	0.289^{***}	0.289^{**}
	(0.094)	(0.111)	(0.105)	(0.128)
wordcount	−2.766e-04	$-8.209\text{e-}04^{*}$	−5.458e-04	$-1.066\text{e-}03^{**}$
	(3.835e-04)	(4.547e-04)	(4.325e-04)	(5.262e-04)
docvocab	−6.082e-05	$2.141\text{e-}03^{**}$	8.013e-04	$1.965\text{e-}03^{*}$
	(8.674e-04)	(1.028e-03)	(9.781e-04)	(1.190e-03)
wordlength	-0.049^{*}	0.016	−0.051	−0.025
	(0.029)	(0.034)	(0.033)	(0.04)
sentlength	1.363e-03	$3.447\text{e-}03^{**}$	1.495e-03	−3.082e-04
	(1.288e-03)	(1.527e-03)	(1.453e-03)	(1.768e-03)
share_repeats	0.206^{*}	−0.1	0.083	0.196
	(0.113)	(0.134)	(0.128)	(0.156)
lexical	0.013	-0.235^{*}	−7.249e-03	−0.073
	(0.102)	(0.121)	(0.115)	(0.14)
colemanliau	0.013^{***}	−3.327e-03	0.014^{***}	6.160e-03
	(4.091e-03)	(4.850e-03)	(4.613e-03)	(5.612e-03)
Observations	500	500	500	500
R^2	0.077	0.055	0.068	0.042
Adjusted R^2	0.064	0.042	0.054	0.029
F Statistic (df = 7; 492)	5.905^{***}	4.107^{***}	5.104^{***}	3.101^{***}
Note:				*p<0.1; **p<0.05; ***p<0.01

TABLE 7.9: Regression output for *dogfood* data set.

	Dependent variable:			
	LDA	SC-LDA	uLDA	SC-uLDA
Constant	0.188***	0.303***	0.311***	0.394***
	(0.067)	(0.084)	(0.077)	(0.099)
wordcount	−1.003e-03	−7.015e-04	1.684e-03*	−5.873e-04
	(8.409e-04)	(1.062e-03)	(9.697e-04)	(1.242e-03)
docvocab	2.154e-03	2.326e-03	−2.483e-03	1.717e-03
	(1.476e-03)	(1.864e-03)	(1.702e-03)	(2.179e-03)
wordlength	1.631e-03	−0.019	−0.013	1.269e-03
	(0.02)	(0.025)	(0.023)	(0.029)
sentlength	1.228e-03	5.510e-04	−2.396e-04	−3.714e-03**
	(1.259e-03)	(1.590e-03)	(1.452e-03)	(1.859e-03)
share_repeats	−0.263**	−0.317**	−0.329***	−0.38**
	(0.108)	(0.136)	(0.124)	(0.159)
lexical	−0.148**	−0.162*	−0.123	−0.248**
	(0.067)	(0.084)	(0.077)	(0.098)
colemanliau	3.425e-03	6.726e-03**	4.381e-03	3.467e-03
	(2.525e-03)	(3.188e-03)	(2.911e-03)	(3.727e-03)
Observations	982	982	982	982
R^2	0.028	0.032	0.017	0.022
Adjusted R^2	0.021	0.025	0.01	0.015
F Statistic (df = 7; 974)	3.939***	4.654***	2.421**	3.177***
Note:				*p<0.1; **p<0.05; ***p<0.01

TABLE 7.10: Regression output for *Brexit* data set.

	Dependent variable:			
	LDA	SC-LDA	uLDA	SC-uLDA
Constant	-3.258^{**}	2.092	-3.548^{*}	0.48
	(1.429)	(2.356)	(2.002)	(2.54)
wordcount	−1.115e-04	8.744e-06	−2.714e-05	1.248e-04
	(7.010e-05)	(1.156e-04)	(9.823e-05)	(1.247e-04)
docvocab	1.086e-04	−2.264e-04	8.725e-05	−5.538e-04
	(2.447e-04)	(4.036e-04)	(3.429e-04)	(4.351e-04)
wordlength	1.209^{***}	−0.423	1.143^{*}	0.097
	(0.444)	(0.733)	(0.622)	(0.79)
sentlength	$9.567\text{e-}03^{*}$	−5.389e-03	0.015^{**}	−8.651e-03
	(5.198e-03)	(8.574e-03)	(7.284e-03)	(9.244e-03)
share_repeats	0.414^{**}	−0.116	0.243	0.046
	(0.163)	(0.27)	(0.229)	(0.291)
lexical	-1.069^{***}	−0.852	0.558	−0.633
	(0.344)	(0.568)	(0.482)	(0.612)
colemanliau	-0.207^{***}	0.074	-0.229^{**}	−0.018
	(0.074)	(0.121)	(0.103)	(0.131)
Observations	168	168	168	168
R^2	0.33	0.055	0.146	0.105
Adjusted R^2	0.3	0.013	0.108	0.066
F Statistic (df = 7; 160)	11.235^{***}	1.32	3.893^{***}	2.676^{**}
Note:				*p<0.1; **p<0.05; ***p<0.01

TABLE 7.11: Regression output for theta shares on *tent* data set.

	Dependent variable:			
	LDA	SC-LDA	uLDA	SC-uLDA
Constant	−0.242*	0.317**	0.575***	−0.216**
topic1	0.049	−0.263	−0.524***	0.651**
topic2	0.596***	−0.663**	−0.722***	0.456**
topic3	0.206	−0.388*	−0.488***	0.341*
topic4	0.104	0.315	−0.519***	0.163
topic5	0.344**	0.81***	−0.152	0.455**
topic6	0.163	−0.502*	−0.696***	−0.236
topic7	0.461***	1.17***	−0.434***	0.023
topic8	0.164	0.423*	−0.011	0.245
topic9	0.734***	−0.486	−0.733***	0.081
topic10	0.107	−0.378*	−0.67***	2.202***
topic11	0.073	−0.476*	0.521***	0.171
topic12	1.159***	−0.388*	−0.8***	0.352*
topic13	1.275***	−0.334	−0.645***	1.07***
topic14	0.691***	−0.47*	−0.787***	−0.062
topic15	0.436**	−0.526**	−0.685***	−0.24
topic16	1.082***	−0.429**	−0.206**	−0.118
topic17	0.177	−0.416**	−0.594***	0.098
topic18	0.249	0.359	−0.668***	0.883***
topic19	0.207	−0.373	−0.587***	0.039
topic20	0.388**	−0.469**	0.305***	0.339**
topic21	8.727e-03	−0.426*	−0.214*	1.317***
topic22	0.299**	1.498***	−0.695***	−0.178
topic23	0.86***	−0.565**	−0.565***	0.599***
topic24	0.113	−0.493**	−0.536***	0.266
topic25	0.236	0.257	0.546***	1.736***
topic26	0.15	0.671***	−0.587***	0.995***
topic27	0.167	−0.561***	−0.746***	0.094
Observations	500	500	500	500
R^2	0.582	0.371	0.633	0.458
Adjusted R^2	0.558	0.335	0.612	0.427
F Statistic (df = 27; 472)	24.329***	10.294***	30.119***	14.75***

Note: *p<0.1; **p<0.05; ***p<0.01

Important: Since the models are estimated independently, there is no association between individual topics. This means that topic 1 of LDA can differ from topic 1 in SC-LDA or uLDA. Comparing one row of models does not provide useful information.

TABLE 7.12: Regression output for theta shares on *dogfood* data set.

	Dependent variable:			
	LDA	SC-LDA	uLDA	SC-uLDA
Constant	0.429***	0.79***	0.547***	−0.219***
topic1	−0.091	−0.955***	−0.678***	−0.088
topic2	−0.318***	−0.698***	−0.589***	0.197
topic3	−0.362***	−0.975***	−0.647***	0.466***
topic4	0.265**	−0.22	−0.127	0.258
topic5	−0.607***	−0.526***	−0.651***	0.352**
topic6	−0.616***	−0.416***	−0.462***	−0.061
topic7	0.29***	−1.028***	−0.407**	−1.029e-03
topic8	−0.337***	−0.453***	−0.664***	0.465***
topic9	−0.024	−0.791***	−0.374***	0.495***
topic10	−0.31***	−0.694***	−0.409***	0.77***
topic11	0.378***	−0.565***	−0.324**	0.449**
topic12	−0.49***	0.369***	0.511***	0.017
topic13	−0.432***	−1.127***	0.031	−0.034
topic14	−0.446***	−0.8***	−0.028	1.397***
topic15	−0.607***	−1.005***	−0.442***	1.648***
topic16	−0.558***	−0.966***	−0.344**	1.474***
topic17	0.258***	−1.068***	−0.624***	0.824***
topic18	−0.437***	0.377***	−0.653***	1.877***
topic19	−0.612***	−0.775***	−0.361***	0.141
topic20	−0.305***	−0.232	−0.379***	−0.048
topic21	−0.406***	−0.797***	−0.693***	0.507***
topic22	−0.343***	−0.556***	−0.42***	−0.171
topic23	−0.477***	−0.92***	−0.698***	−0.117
Observations	982	982	982	982
R^2	0.375	0.328	0.354	0.408
Adjusted R^2	0.36	0.312	0.339	0.394
F Statistic (df = 23; 958)	24.947***	20.311***	22.859***	28.682***

Note: *p<0.1; **p<0.05; ***p<0.01

Important: Since the models are estimated independently, there is no association between individual topics. This means that topic 1 of LDA can differ from topic 1 in SC-LDA or uLDA. Comparing one row of models does not provide useful information.

TABLE 7.13: Regression output for theta shares on *Brexit* data set.

	Dependent variable:			
	LDA	SC-LDA	uLDA	SC-uLDA
Constant	0.052	−0.057	0.024	0.268***
topic1	−0.21*	0.339*	−0.012	−0.04
topic2	0.425***	0.081	0.17***	−0.241*
topic3	−0.053	0.122	0.306***	−0.035
topic4	0.19	0.35**	−0.011	−0.016
topic5	−0.101	0.226	0.458***	−0.289**
topic6	−0.154	0.609***	0.142***	−0.032
topic7	0.212	0.311	0.125**	−0.239
topic8	−0.228**	0.043	0.16***	0.161
topic9	−0.131	0.575***	0.236***	0.136
Observations	168	168	168	168
R^2	0.248	0.171	0.38	0.129
Adjusted R^2	0.206	0.124	0.345	0.08
F Statistic (df = 9; 158)	5.805***	3.633***	10.775***	2.61***

Note: *p<0.1; **p<0.05; ***p<0.01

Important: Since the models are estimated independently, there is no association between individual topics. This means that topic 1 of LDA can differ from topic 1 in SC-LDA or uLDA. Comparing one row of models does not provide useful information.

Chapter 8

Discussion

The results presented in chapter 7 confirmed that the uLDA model provides a valuable extension to existing topic models. Since this thesis both introduced the concept of ubiquitous terms and proposed a model based on that concept, the following chapter will contextualize and assess the results of both strains. Additionally, possible implications of the findings presented in the results chapter will be provided.

The first section is concerned with the concept of ubiquitous terms as discussed in section 4.2 and the naturally occurring non-topic examined in section 7.1. This will illustrate how the theory is backed up by the model-free analysis and further strengthened by the phenomenon of an "natural non-topic."

This is followed by a section on implications of the topic model results, which re-emphasizes the benefit of incorporating stop words into the model. Finally, an extensive discussion on the human-based experiment will review both the method used and its results while also suggesting improvements for future experiments.

8.1 Ubiquitous Terms and the Natural Non-Topic

The extended descriptive analysis in section 5.3 has shown that ubiquitous terms seem to be a valid concept supported by empirical evidence. The frequency analysis demonstrated that some terms are in fact ubiquitous, meaning that they appear across all documents in a similar, high frequency. It was also shown that these terms have a considerable albeit incomplete overlap with common stop words.

In the context of topic models, this means that those terms cannot be clearly assigned to a certain topic since the topic assignment is informed by within-document co-occurrence. Terms that appear in near-constant frequency across documents display no variation in co-occurrence, and therefore there is little information on which topic these words should be assigned to. This is supported by the fact that topic model results often contain stop words in all resulting topics if these terms are not removed beforehand.

The novel solution provided by this thesis is to absorb all these terms in a separate probability vector that specifically includes words with a near-constant frequency across all documents. Since these terms are not assigned to any distinct topic, this collection of word probabilities was dubbed "non-topic." Due to the ubiquitous nature of the terms within this non-topic, the implementation developed in section 6.2.1 used a corpus-wide constant probability for a term to be allocated to this new collection of terms.

This received further support from the results in section 7.1, where it was demonstrated that non-topic-like topics could appear naturally in LDA, given specific settings. Although not intended, this kind of topic shows all the characteristics of ubiquitous terms.

1. It appears in high frequency across all documents.
2. It has only little variance in frequency across documents.
3. It contains mainly typical function words or non-informative terms.

The sub-study in Section 6.1 was conducted to test the hypothesis that the average document size within a corpus is contributing to the appearance of such a "natural non-topic." When juxtaposing the results in Section 7.1 with the criteria above, it can be confirmed that document size does play into this phenomenon. Figure 7.1 shows that, for the more rigid sub-samples (sub-figures C and E), one topic is selected far more frequently than others. Figure 7.2 confirms the second point in showing that the dominant topic is evenly distributed across all documents, although the effect is more pronounced for the tent data set. The third point is backed by Figure 7.3, which displays the most prominent word of the respective topic.

However, it seems that the emergence of a natural non-topic in LDA results can not be explained solely by the average document size. As the comparison with sub-samples from the Reuters data has shown, other

factors might be relevant, such as overall corpus size or heterogeneity in topics. Nevertheless, it is clear that this effect is not a one-time anomaly unique to the *Brexit* data set.

This means that even a standard LDA finds enough evidence in the provided data to identify such non-topic. The problem with relying on a natural non-topic instead of incorporating the concept into the model itself lies with the model definition. It is not guaranteed that such topics would still appear if LDA is modified. In fact, the SC-LDA model can not produce such topics due to its sentence constraint. The issue that topics can only be assigned to whole sentences leads inevitably to ubiquitous terms being distributed almost evenly across topics. Therefore, if the goal is to utilize non-topics, they have to be included in the model.

8.2 Comparing Topic Models

The model fit measures presented in section 7.2.2 show that topic models that include ubiquitous terms better fit the holdout data than the traditional counterparts. Additionally, the perplexity values displayed in Table 7.5 show that regarding model fit, including the ubiquitous terms in the model clearly provides a more sophisticated alternative to established pre-processing methods.

It should be stressed that this validates the central hypothesis of this thesis, which is that including these terms would be beneficial to model performance. The new model outperforms standard LDA on both the full and the stop words free data sets. This is a strong indication that the information conveyed by typical stop words is indeed relevant to topic models.

The results also confirm the findings in Büschken and Allenby (2016) by showing better fit measures for SC-LDA. However, it stands out that the SC-uLDA model provides a slightly worse fit than simple uLDA. This is a surprising result since the advantage is switched for the standard models.

A possible explanation could be that the strict boundaries imposed by the sentence structures within the SC-uLDA do not allow the uLDA to fully benefit from the increased flexibility that the non-topic provides. Since changing a topic mid-sentence is not allowed in this model, some model fit is inevitably lost on multi-topical sentences. This effect could

possibly be mitigated by relaxing the strict boundaries. Overall, the fit measures reinforce the confidence in the new model extensions.

Besides the fit measures, the results chapter included examples of topic word clouds. These figures display topic-related terms in different sizes dependent on their importance within that topic, measured by their ϕ-values. Although comparing word clouds is a qualitative approach that inherently is prone to subjectivity and possibly bias, the figures should be taken as an indication on the possibilities that uLDA models provide for the interpretation of topics.

The examples in Figures 7.6, 7.7, and 7.8 show that ubiquitous terms models tend to reduce the prevalence of stop words, although not in equal scope across all cases. Sometimes the topics still contain stop words. However, the importance of these terms in relation to topic-specific terms is clearly reduced. This downranking of non-topic terms would allow for a more straightforward interpretation by users.

In conclusion, classic fit measures confirm that the proposed topic models clearly benefit from including ubiquitous terms. The fact that these model consistently score higher than their "traditional" counterparts shows that the additional information provides a substantial benefit.

8.3 Human-Based Experiment

The experiment presented in section 6.3 provided a novel approach to evaluating topic model results. Since the literature offers no indication of a similar experiment being conducted, an extensive discussion of the method and its results are needed. The first part of this section is concerned with evaluating the results, followed by implications for both topic models and study design.

The experiment provided topic labels for each term within a document. This enables not only direct comparison with topic model outputs but also provides additional information about the labeling behavior of the participants. As shown in table 7.3, the provided labels often occur in runs, i.e., consecutive words with the same topic assignment. This means that there is a tendency to label chunks of words with the same topic, often bounded by sentence markers or other grammatical stop points such as conjunctions. Since no words were removed for the task, these chunks would naturally include function words such as "and," "the,"

or "to." This either means that those terms are relevant to the topic, which is unlikely for a word such as "the," or that the worker glosses over function words, ignoring them when making the topic label decision. In the first case, the term should not be removed. But in the latter case, it should also not be included in the topic since it only happens to be in the chunk by chance. This can be seen as another case for a stochastic allocation of non-topic words as it is implemented in uLDA.

The phenomenon of "topic runs" was examined further by looking at the change points provided in Table 7.4. The results showed that in 76.7 percent of all cases, topic changes appeared at sentence boundaries such as full stops or question marks. When other reasonable markers such as brackets, commas, and conjunctions are included, a total of 93.7% of change points can be explained.

Overall this suggests that a sentence constraint can help identify the topic runs labeled during the experiment. However, the limitation on sentence boundaries can be too strict in some cases. This might be an interesting starting point for developing the model further.

8.3.1 Model Implications

Based on the labels provided by this experiment, hit rates were calculated in section 7.3. These scores can provide some information about model performance. The average hit rates for labeled documents were calculated as described in equation 6.21. The results were classified by model and data set and are shown in table 7.6.

When looking at the results based on the human-based tagging, it should be noted that directly comparing the hit rate percentage values would be improper. Random allocation of topics would lead to an expected hit rate of $\frac{1}{T}$, with different T for the different data sets. Therefore, the model hit rates can be transformed into "hit rate lift scores," which indicate by what factor the model outperforms the respective baseline.

Table 8.1 shows that this leads to the consumer review corpora performing distinctly better than the editorial corpus. This could be related to the variance in corpus characteristics. The model results hint to differences in the language between the two types of data sets, reviews and editorials. One hint for this are the estimated values of δ, which represents the share of topic-related words in a corpus. Its posterior mean lies between 0.49 and 0.63 for the consumer review data sets but drops as far

Table 8.1: Table of hit rate lift compared to random baseline of $\frac{1}{T}$.

	LDA	SC-LDA	Ubi-LDA	Ubi-SC-LDA
tent	4.4	5.7	5.0	6.6
dogfood	3.3	4.4	4.0	5.5
Brexit	1.6	2.4	1.7	2.2

as 0.26 for the *Brexit* data set. In other words, almost three-quarters of the terms in the editorial data set are classified as "not topic-related" by the SC-uLDA model. This might be an indication that the documents have little variety in vocabulary and contain filler words that do not contribute to a specific topic. These discrepancies in language probably contribute to the differences in hit rate performances.

Another possible explanation might be the difference in the number of topics. It is possible that the 10 topics that were used to tag the *Brexit* data set lead to broader, less precise topics that are more difficult to reproduce for a topic model. Unfortunately, without obtaining additional topic labels from human workers, this cannot be tested. However, there is evidence that suggests that more complex texts might have influenced topic labels. Table 8.2 shows the share of human-tagged within-document topic changes that appear at the start or end of a new sentence. Although this is generally very high with values of 66% and 79% for the consumer reviews, the *Brexit* data set stands out with over 97% of all within-document topic changes coinciding with sentence boundaries. This might be a part of the explanation why the standard SC-LDA performs better on this data set compared to the other two.

Table 8.2: Share of topic changes that appear at the start or end of a sentence. Only topic changes that appear within a document were considered.

tent data set	*dogfood* data set	*Brexit* data set
78.9%	66.1%	97.1%

In summary, the results of the human-based tagging experiment show that more complex models outperform standard LDA. Especially when looking at consumer reviews, ubiquitous term models consistently outperform their traditional counterparts. This suggests that the proposed extension brings the LDA closer to mimicking human language perception when defining topics. Unfortunately, the human codings do not contain flags for non-topic terms, so validating the non-topic ψ is not possible based on the available data. This might be an interesting addition for future topic labeling projects.

A regression analysis with document characteristics as independent variables was performed in 7.3 to identify possible drivers of hit rates. Unfortunately, the results of this analysis are rather inconclusive. Although there are significant coefficients in almost all regression models, the influence seems to differ from model to model and even from data set to data set. Some of the more common regressors are the number of words per document and the document vocabulary. The results should be interpreted with caution since the adjusted coefficient of determination (R^2) is very low, suggesting that the variables can only explain a small fraction of the variance in hit rates. This means that although there are indications that the language complexity plays into model performance, its effect size is most likely negligible.

If anything can be read from these outputs, it is that the model most affected by these parameters is LDA, while other model performances, especially for ubiquitous term based models, can barely be explained by the examined characteristics. Although no substantial effect was found, this should still be considered a positive outcome since it also means that topic model performance is not impaired by variation in document structure.

It should also be mentioned that the model implementations were optimized for readability and not for speed. Since only a minimum of terms were removed, the estimation process took a considerable amount of time for both standard LDA models and ubiquitous terms extensions. This might be an issue when implementing these models in a business context but is not addressed here since it is outside the scope of this thesis.

8.3.2 Study Design Implications

Since the experiment presented in section 6.3.1 is a novel approach, there is a lot to be learned for future instances of comparable studies. Certain topics produce very low hit rates, while others are matched quite well. A reason for this might be that human coders can interpret context and understand intent while topic models can only access the term co-occurrence since they have no concept of meaning. The fact that the *Brexit* data set shows lower variation in hit rates could also signify that the optimal topic count could actually be lower for the *tent* and *dogfood* data sets.

A closer look at the actual topics that achieve particularly high or low hit rates is given in tables 8.3 and 8.4. The "best matched" topics, i.e., the topics that contained the most topic hits on average, are usually about distinct characteristics that can easily be differentiated from others. Examples for this are the customer service (*tent* data set) or the container the food comes in (*dogfood* dataset). On the other hand, poorly performing topics are either very general, such as the "historical/general background" topic in the *Brexit* data set, or relatively small such as the "quantity" (*dogfood*) or the "innovativity" topic (*tent*). Both were assigned only 31 and 8 times in total, respectively. There is also an obvious problem with the "innovativity"[1] topic: it is so rare that its defining term, "innovation," seems to be left below the threshold of rare terms and got removed [2].

These results might give some hints for further development of the experimental design. It could be argued that very small topics should be avoided if possible and that overly broad topics might better be divided into more concrete ones.

When looking at the experiment results as a whole from all the angles presented within this thesis, several starting points for future improvements can be identified. The main issues that will be addressed here are subjectivity, topic set size, topic differentiation, ubiquitous terms, and practical implementation.

Subjectivity The first adjustment is aimed at the issue that language perception is subjective. The resulting labels do most likely depend on

[1] This label is a direct quote from the experiment results.

[2] The respective topic consisted only of the sentence "The tent is a great innovation in large tents."

TABLE 8.3: Two of the highest scoring topics across models by data set.

The topics are represented by the 10 most likely terms according to the posterior distribution of the respective ϕ.

Brexit		dogfood		tent	
Brexit	Brexit	bag	small	coleman	bugs
johnson	deal	holes	size	pole	area
deal	johnson	bags	dog	replacement	mosquitoes
eu	eu	box	treats	customer	screened
party	britain	just	dogs	tent	tent
labour	european	arrived	break	dont	able
britain	british	food	smaller	tried	annoying
people	minister	hole	broken	warranty	dont
leave	union	im	like	called	great
british	parliament	long	pieces	get	keep

TABLE 8.4: Two of the lowest scoring topics across models by data set.

The topics are represented by the 10 most likely terms according to the posterior distribution of the respective ϕ.

Brexit		dogfood		tent	
Brexit	Brexit	expiration	bag	material	great
johnson	eu	date	pound	tent	large
eu	johnson	away	30	better	tent
deal	deal	bag	6	materials	tents
labour	britain	dates	largest	well	
britain	british	expired	ordered	50	
vote	ireland	good	paid	6	
party	parliament	guarantee	quantity	andor	
parliament	now	product	received	breezes	
british	party	store	yet	canvas	

prior knowledge and the workers' reading comprehension skills. A larger pool of participants could better ensure reproducibility by balancing the subjective influences. For this, two main approaches seem reasonable. The first would be to employ H workers who individually label documents according to the given protocol, resulting in H topic sets of possibly varying size and H sets of topic labels. These results can then be taken to calculate H different hit rates, of which the arithmetic mean could provide an overall score for each model. An additional benefit of this lies in informing the researcher about the distribution of possible hit rates.

Another approach to this issue would be to combine several participants into a group that jointly decide on topic set and document labels. This could reduce the subjectivity in the resulting assignments of z, depending on the number of persons and possibly also on group dynamics. However, there might be a loss of efficiency in tagging due to the coordination costs. Both solutions mentioned above can also be combined to form several groups or mixed to only define the topic sets within the group while keeping the labeling process on an individual level.

Unfortunately, all suggestions require enrolling additional workers and therefore will inevitably lead to higher costs.

Topic Set Size The second suggestion for improving the experimental design is related to topic set size. Note that topic set size, i.e., the number of defined topics, is directly related to topic size, i.e., the number of tokens per topic, since a larger topic set leads to an smaller average topic size and also gives more room for niche topics consisting of relatively few terms. In the presented form, there is no mechanism to control for topic set size within the experiment, which could lead to the human coders "overfitting" the data. It also seems that workers were tempted to "fine-tune" topics when new documents presented them with unexpected content.

Therefore, it might be advisable to implement a two-stage process where participants are given only a subset of the data along with the task to define a set of underlying topics. Upon completion, the rest of the data is provided with the requirement to adhere to the previously defined topic set. This procedure is arguably very similar to the protocol presented in this study. Nevertheless, the stricter implementation would allow for better control of the results by eliminating the ability to gloss over the

results and compelling the workers to possibly make "incongruous" topic assignments for some fringe cases.

This issue might also be mitigated when the group approach is implemented as described above since the majority of participants would have to be convinced that the respective topics are a necessary part of the topic sets.

Topic Differentiation A matter closely intertwined with the issue of *topic size* is topic differentiation. This close relationship is evident when considering that a larger topic set size allows for more specific topics and vice versa. Topic differentiation also poses an especially complex problem since there is no "right" direction to take. Allowing for more specific topics could lead to "overfitting"-like behavior and might result in very small topics that are hardly detectable by topic models, which rely on variance in co-occurrence. On the other hand, discouraging narrow topics could lead to very broad topics that are not easily differentiated from others.

It might be a reasonable approach to encourage a balance in topic sizes across the topic set. However, this should probably not be a hard criterion since in some cases, there might be good arguments for small, precise topics regarding specific issues within a corpus. One approach might be to define some very low criteria that bring fringe cases to the workers' attention without imposing hard boundaries. Examples for this are topics that appear in less than 0.1% of documents or topics that make up more than 4 or 5 times of the token than average. The convenience of the latter example is that this can be achieved in two ways. Either reduce the respective topic's size by making it more specific or increase the average topic size by reducing the total number of topics.

However, it should be stressed again that too strict rules might be counter-productive and also not necessary. The first step should be to bring the issue to the workers' attention and hope the human mind figures out a suitable middle ground.

Ubiquitous Terms Another point mentioned earlier is that with the current protocol, there is no distinction between topic-related and topic-irrelevant terms, i.e., ubiquitous terms in the sense of uLDA. This lack of differentiation makes it impossible to evaluate the quality of the non-topic, which leads to the respective terms being omitted in the hit rate scoring step.

Obtaining this classification would be useful since it could provide further evaluation for the concept of ubiquitous terms. Moreover, knowing which terms inform topics and which do not would be an interesting point of research even without its relation to the uLDA model. It could provide insights into human language perception in general and help develop these models further.

Practical Implementation The last point worth mentioning is concerned with the practical implementation of the experiment. The first trials were conducted by annotating text files with numbers ranging from 1 to T. This procedure was both time-consuming and susceptible to slip-ups. It might be helpful to employ a computer-assisted labeling method that can ensure hard boundaries, such as surjective mapping of tokens to topics, while increasing awareness for soft boundaries by, for instance, highlighting topics that only contain a few words. Making the text immutable would also ensure consistency in that area. The ability to click on tokens in order to assign topics could also decrease the time cost. Finally, a software-assisted solution would also reduce the room for subjective interpretation of the experiment protocol in general, which could reduce miscommunication and prevent issues that might not yet be on the radar.

The main caveat with this solution would be that providing too much feedback during the ongoing labeling process could have unintended consequences. Participants could feel compelled to adhere to suggestions or hints given by the tool, which would transmute soft boundaries into hard boundaries. To ensure that the resulting labels are still based on human perception and therefore can be considered as "ground truth" for the model, the feedback to the worker should probably be kept to a minimum. The exact design should itself be tested in experiments in order to find an ideal compromise.

Summary Although there is still room for improvement in the experiment process, the results can provide some form of performance measure for the models. As results have shown, the models do outperform their baseline of a randomly generated allocation of topics significantly, which is a clear sign that human-based topic labels are worth closer examination in the future.

The analyses of the results of this short run have provided several starting points for future improvement that should be considered when

similar experiments are conducted. In hindsight, the protocol could be improved in some points in order to bring the results closer to the goal of the minimal set of topics to fully describe the corpus content. Also, some clarifications could facilitate the labeling process for the participants. Additionally to conceptual modifications, introducing some form of digital assistance could reduce cost by reducing time and error rate.

Nevertheless, it should be kept in mind that there is a risk of over-engineering the labeling process. If too many rules and regulations are provided, the result might stray away from creating human-perceived topics towards human-executed topic modeling. This possibility should always be kept in mind when imposing new restrictions or guidelines. In general, it could be advisable to avoid examining topic model results on a given corpus before conducting the experiment in order to avoid any subconscious bias when designing the protocol.

Even though the study design might not be fully optimized yet, the results have proven to be an excellent complement to established methods. The experiment in this thesis provides some groundwork for the field of human-based topics. After the successful proof of concept, there is still some room for improvement. If done correctly, human-based topics could provide an interesting addition to current evaluation methods, especially when examining new topic model extensions in an academic context.

Part IV

Conclusion

The presented thesis aimed to narrow the gap between natural language and topic models by reducing pre-processing. This objective was approached from two directions. First, from the topic model perspective, an alternative concept to the classic understanding of stop words was introduced in the form of *ubiquitous terms*. These findings were used to formulate the uLDA model. And second, from the natural language perspective, an experiment was conducted to better understand how humans perceive topics and topic labels. The results of this course of action will be summarized as answers to the research questions formed in chapter 2.

1. **Does removing stop words before model estimation change the interaction between LDA and the given data**
   The analyses in section 5.3 have shown that stop words contribute to the topic formation during LDA in the sense that the respective terms are not distributed equally across topics. Some stop words are grouped into single topics, while others frequently appear across the whole topic set. This diversity suggests that including stop words in topic modeling elevates the information level and hence could provide a better fit. Furthermore, the occurrence of "natural non-topics" as examined in section 6.1 has shown that, under specific circumstances, terms not relevant to any specific topic are grouped together, an effect that was used in the implementation of uLDA in section 6.2.1.

2. **Can LDA be extended in order to harness possible interaction effects? Could this allow for a reduced pre-processing that would warrant the input data to be closer to natural language?**
   The evaluation of uLDA has shown that the newly introduced non-topic does, in fact, absorb some of the typical stop words while allowing some of them to still appear prominently in certain topics. This means that uLDA can use the characteristics of ubiquitous terms and separate them from topic-related terms. Not removing ubiquitous terms does preserve more of the original language of a corpus, and uLDA can utilize this information that is now kept in the data.

3. **Can reference values for human-perceived topics be obtained in an experimental setting? How do humans perceive topics in the context of such an experiment?**

This thesis proposed an experiment for generating human-based topic labels on a word level. The introduced protocol was mainly aimed at mimicking topic model behavior by generating a set of T topics and providing a surjective mapping from any model output to these topics for all tokens within a corpus. The experiment offered interesting data that was used to evaluate all estimated models by matching the human topics against the model topics and calculating an average hit rate for each model. A descriptive analysis of the resulting human-based topics showed that humans seem to perceive topics in chunks rather than on a word level, which is an intriguing finding regarding possible future model extensions.

4. **Does such an extension actually provide a better model for human-perceived topics?**
   The evaluation showed that models based on ubiquitous terms outperform standard LDA across all data sets when measuring the experiment hit rate. In one case, SC-LDA showed slightly better values than the uLDA variants. Reasons for this might be differences in the document characteristics; however, this phenomenon should probably be examined closer in future research. In most cases, uLDA and SC-uLDA show a better fit to both data, measured by out-of-sample log-likelihood, and human coding, measured by hit rate.

Based on these results, several paths for future research can be identified. Concerning the uLDA model, the most obvious step would be to scale up the data. This approach could include using more data sets with a greater variety in characteristics, just using larger data sets, or a combination of both. It could provide a better understanding of this model's advisable applications and provide an explanation for the unexpectedly high performance of SC-LDA in the *Brexit* data set.

For the experiment, section 8.3.2 provided some thoughts on improving the protocol. The most straightforward next step here would be expanding the setting to include more workers, which mainly poses cost restrictions. Another beneficial adjustment could be introducing an assisting software tool that guides the process and thus ensures quality control. And finally, with the variety of tweaks suggested in the discussion part of this thesis, even deeper insight could be gained by evaluating different variations of the experiment to assess the effects of each suggested modification.

Finally, regarding the proposed model, there are a variety of possible approaches to take from here. The simple uLDA model and the SC-uLDA extension have both been demonstrated to provide better results on tasks directly linked to natural language modeling. As was discussed in section 6.2.1, the elegance of the ubiquitous term extension lends itself to being incorporated in many other topic models. Combining this concept with other existing topic model extensions is clearly an appealing issue to be examined.

Additionally, the descriptive analysis of human-based topic labels in section 7.3 points in a particular direction for future research. The fact that topic labels appear in blocks and do not seem to change after every word is an indication that topic models that include such a structure might perform better when trying to approximate human perception. The SC-uLDA model already represents the first step in that direction. However, the sentence boundaries seem a bit too strict when considering that shorter document corpora had only about roughly 70% of topic change points located right next to punctuation. A more flexible sizing of these topic runs might be a promising next step.

In conclusion, this thesis provided groundwork on how to approach natural language from a topic model standpoint. The proposed model extensions have demonstrated promising performance while also incorporating typical stop words that would otherwise have been removed. Moreover, the conducted experiment introduces a new approach to topic model evaluation focusing on human topic perception. Both these results encourage further research in the field with the hopes of bringing topic models even closer towards natural language.

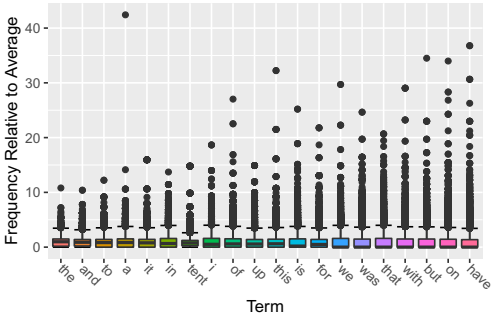
Appendix

Appendix A

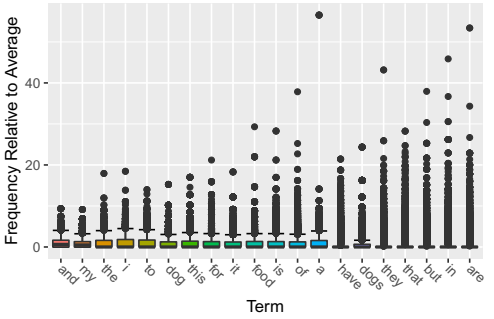
Empirical Analysis

Model Free Analysis

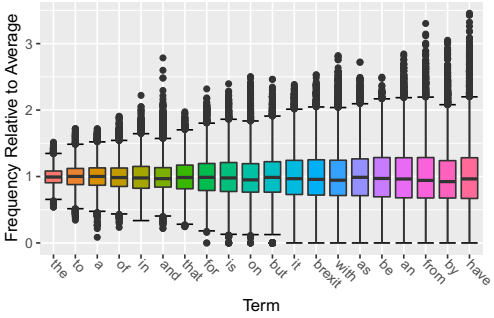
This section contains additional plots to complement those of section 5.3. The consumer review data sets are used as is (in contrast to the plots in the thesis), while the *brexit* data set was prepared with repeated sampling as described in the main matter. Figure A.1 displays these results. The main finding is not changed by this, although the visualization differs. Slight changes in the ordering occur, however this does not impact the conclusions drawn from this analysis.



(A) Tents



(B) Dogfood



(C) Brexit

FIGURE A.1: Frequency distributions for the different data sets.

Appendix B

Derivations

B.1 Derivation of uLDA Gibbs Sampling

TABLE B.1: Notation for uLDA Gibbs sampling.

notation	description
Data	
w	word
W	Number of words in the corpus
Latent Variables	
z	topic assignment
θ	topic distribution
ϕ	word distribution for every topic
ψ	word distribution for "non-topic"
τ	binary topic/non-topic indicator
δ	a priori probability of $\tau = 1$
Fixed Priors	
α	prior to θ
β	prior to ϕ
Other	
C^{W0}	count vector for non-topic assignments per unique token
C^{DT}	count matrix for topic assignments per document

The LDA model by Blei, Ng, and Jordan (2003) is at its core an hierarchical Bayes model. For this type of model, parameter estimation is usually approached via posterior inference, which depends on calculating

the posterior distribution of all model parameters. For standard LDA, this was explained briefly in section 3.1. This process can be adapted for uLDA, in which case the joint posterior distribution of all model parameters is

$$p(\theta, \phi, \psi, \delta, \tau, z \mid w, \alpha, \beta) = \frac{p(\theta, \phi, \psi, \delta, \tau, z, w \mid \alpha, \beta)}{p(w \mid \alpha, \beta)}, \tag{B.1}$$

The Gibbs sampling method uses full conditional distributions to sample each unknown parameter from a distribution of that specific parameter conditional on all others. For uLDA this means a full conditional is needed for θ, δ, ϕ, ψ, τ, and z. The process is simplified drastically when considering that draws on word level are i.i.d., which leads to the solution that for all $\{w_i \mid \tau_i = 1\}$, the sampling of z_i is identical to LDA. Since δ is irrelevant to the draw of z_i if $\tau_i = 1$ is given. The same holds true for ψ, which is only relevant for non-topic terms. Note that the collapsed Gibbs sampler introduced in equation 3.5 is not applicable here, which slightly increases the time to convergence.

$$p(z_i \mid \theta, \tau_i, w_i, \phi, \psi, \delta) \propto p(w \mid \phi_{z_i}) \cdot p(z_i \mid \theta_{d_i}) \tag{B.2}$$

which is, in full Gibbs:

$$p(z_i = t \mid w_i, \phi, \theta) \propto \phi_{w_i,t} \cdot \theta_{d_i,t} \tag{B.3}$$

and consequently:

$$p(z_i = t \mid w_i, \phi, \theta) \sim Mult\left(\frac{\phi_{w_i} \cdot \theta_{d_i}}{\sum_{t \in T} \phi_{w_i,t} \cdot \theta_{d_i,t}}\right) \tag{B.4}$$

For θ, nothing changes. The only difference is, that the count matrix C^{DT} only includes terms for which τ was sampled as 1. In this case, δ can again be omitted since τ is given, which allows for copying the distribution from LDA:

$$\begin{aligned} p(\theta_d \mid z_d, \tau_d, \alpha) &\propto p(z_d \mid \theta_d) \cdot p(\theta_d \mid \alpha) \\ &\sim Dir(C^{DT} + \alpha) \end{aligned} \tag{B.5}$$

This is also the case for drawing ϕ and ψ. In LDA, ϕ_t is already only dependent on terms with $z_i = t$, which means non-topic terms are

ignored. For ψ, the draw is equivalent to ϕ_t with the exception that the relevant terms are not selected via $z_i = t$, but $\tau_i = 0$.

ϕ:

$$\begin{aligned} p(\phi \mid w, z, \beta, \tau = 1) &\propto p(w \mid \phi_z) \cdot p(\phi \mid \beta) \\ &\sim Dir(C^{WT} + \beta) \end{aligned} \tag{B.6}$$

ψ:

$$\begin{aligned} p(\psi \mid w, z, \gamma, \tau = 0) &\propto p(w \mid \psi) \cdot p(\psi \mid \gamma) \\ &\sim Dir(C^{W0} + \gamma) \end{aligned} \tag{B.7}$$

The first real new distribution comes with the draw of τ. Since this parameter is assumed to be Bernoulli distributed, only the probability of $\tau = 1$ has to be determined, since $p(\tau = 0) = 1 - p(\tau = 1)$. Following Bayes rule, this probability can be split up into likelihood times prior as follows:

$$p(\tau_i \mid w_i, \phi, \psi, \theta, \delta) \propto p(w_i \mid \tau_i, \phi, \psi, \theta, \delta) \cdot p(\tau_i \mid \delta) \tag{B.8}$$

$$\tag{B.9}$$

Using the division of favorable outcomes ($\tau = 1$) by possible outcomes ($\tau = 1, \tau = 0$), this leads to $P(\tau = 1)$ being:

$$\begin{aligned} &p(w_i \mid \tau_i, \phi, \psi, \theta, \delta) \cdot p(\tau_i \mid \delta) \propto \\ &\propto \frac{p(w_i \mid \phi, \theta, \tau_i = 1) \cdot p(\tau_i = 1 \mid \delta)}{p(w_i \mid \phi, \theta, \tau_i = 1) \cdot p(\tau_i = 1 \mid \delta) + p(w_i \mid \psi, \tau_i = 0) \cdot p(\tau_w = 0 \mid \delta)} \end{aligned} \tag{B.10}$$

Luckily, the prior distribution $p(\tau_i \mid \delta)$ was proposed as a Bernoulli distribution, which simplifies that part to:

$$p(\tau = 1 \mid \delta) = \delta \tag{B.11}$$

$$p(\tau = 0 \mid \delta) = 1 - \delta \tag{B.12}$$

This leaves only the likelihood for both cases. For $\tau = 0$ this is simply given by the non-topic distribution:

$$p(w \mid \psi, \tau = 0) \propto \psi^w \tag{B.13}$$

And for $\tau = 1$, the case is identical to the LDA likelihood of w. However, a fixed value for z_i is not always available. If the previous draw of $\tau_i = 0$, z_i has no value. Therefore, z_i is integrated out by summing over all possible values, which leads to:

$$\begin{aligned} p(w \mid \phi, z_i, \theta, \tau_i = 1) &\propto p(w_i \mid \phi_{z_i}, \tau_i = 1) \\ &\propto \phi_{w_i, z_i} \\ \text{integrate z out:} & \\ &\propto \sum_{z_i}^{T} (\phi_{w_i, z_i} \cdot \theta_{d_i, z_i}) \\ &\propto \phi_{w_i} \times \theta_{d_i} \end{aligned} \tag{B.14}$$

Putting all these simplification together leads to the probability of $\tau_i = 1$ of:

$$p(\tau_w = 1 \mid \phi, \psi, \theta, \delta) \propto \frac{\phi_{w_i} \times \theta_d \cdot \delta}{\phi_{w_i} \times \theta_{d_i} \cdot \delta + \psi_{w_i} \cdot (1 - \delta)} \tag{B.15}$$

Finally, the full conditional distribution of δ depends on τ and $\gamma = (\gamma_1, \gamma_2)$. This can be split up into

$$p(\delta \mid \tau, \gamma) \propto p(\tau \mid \delta) \cdot p(\delta \mid \gamma) \tag{B.16}$$

Since $p(\tau \mid \delta)$ is Bernoulli distributed and $p(\delta \mid \gamma)$ was introduced as Beta distributed, this leads to:

$$
\begin{aligned}
p(\delta \mid \tau, \gamma) &\propto \binom{n}{k} \delta^{k} (1-\delta)^{n-k} \cdot \frac{1}{B(\gamma_1, \gamma_2)} \delta^{\gamma_1 - 1} (1-\delta)^{\gamma_2 - 1} \\
&\propto \frac{1}{B(k+\gamma_1, n-k+\gamma_2)} \delta^{k+\gamma_1 - 1} (1-\delta)^{n-k+\gamma_2 - 1} \\
&\sim Beta(k+\gamma_1, n-k+\gamma_2)
\end{aligned}
\tag{B.17}
$$

Since n is the number of draws, it equals to the total number of token in the corpus W. In turn, k is the number of favorable outcomes, in this case draws resulting in $\tau = 1$. This equals to the sum of the count matrix C^{W0} across the whole vocabulary V. Consequently, $n - k$ is equal to the difference between W and the sum of C^{W0}. This leads to the final description of the full conditional of δ as:

$$
p(\delta \mid \tau, \gamma) \sim Beta(\sum_{v}^{V} \left(C_v^{W0}\right) + \gamma_1, \left(W - \sum_{v}^{V} C_v^{W0}\right) + \gamma_2). \tag{B.18}
$$

B.2 Derivation of SC-uLDA Gibbs Sampling

The SC-uLDA model is based on an extension proposed by Büschken and Allenby (2016). A detailed description of all Gibbs sampling procedures for their SC-LDA model can be found in their paper. This section builds on their work in order to provide Gibbs sampling steps for the SC-uLDA model.

For the ubiquitous term extension to the SC-LDA, full conditional distributions are needed for the parameters θ, δ, ϕ, ψ, τ, and z. In contrast to the uLDA model, draws of z_i on a word level are no longer considered independent, since topic assignment is done at a sentence level. The non-topic assignment of τ_i is still separate for each token, however it is now conditional on $z_{d,s}$. The joint posterior distribution of all model parameters can still be described as

$$
p(\theta, \phi, \psi, \delta, \tau, z \mid w, \alpha, \beta) = \frac{p(\theta, \phi, \psi, \delta, \tau, z, w \mid \alpha, \beta)}{p(w \mid \alpha, \beta)},
$$

with the only difference being in the individual sampling steps.

TABLE B.2: Notation for SC-uLDA Gibbs sampling.

notation	description
Data	
w	word
s	sentence
W	Number of words in the corpus
Latent Variables	
z	topic assignment
θ	topic distribution
ϕ	word distribution for every topic
ψ	word distribution for "non-topic"
τ	binary topic/non-topic indicator
δ	a priori probability of $\tau = 1$
Fixed Priors	
α	prior to θ
β	prior to ϕ
Other	
C^{W0}	count vector for non-topic assignments per unique token
C^{DT}	count matrix for topic assignments per document

The first important thing to note here is that some draws do not change from the uLDA model. This is the case for θ (equation B.5), ϕ (B.6), ψ (B.7), and δ (B.18). This leaves the only changes to the draws of τ and z.

For the draw of τ, equation B.10 still holds, but the likelihood of w_i changes. Equation B.14 is modified, since in the case of SC-uLDA, the topic assignment $z_{d,s}$ of the respective sentence is known. Therefore, z is not integrated out which leaves the likelihood at:

$$\begin{aligned} p(w \mid \phi, z_{d,s}, \theta, \tau_i = 1) &\propto p(w_i \mid \phi_{z_{d,s}}, \tau_i = 1) \\ &\propto \phi_{w_i, z_{d,s}} \end{aligned} \tag{B.19}$$

With everything else equal to the uLDA version, this leads to the probability of $\tau_i = \tau_{d,s,m} = 1$ of:

$$p(\tau_w = 1 \mid \phi_{w_i, z_{d,s}}, \psi_{w_i}, \delta) \propto \frac{\phi_{w_i, z_{d,s}} \cdot \delta}{\phi_{w_i, z_{d,s}} \cdot \delta + \psi_{w_i} \cdot (1 - \delta)} \tag{B.20}$$

In turn, the draw of $z_{d,s}$ is dependent on τ since only topic-related words are used to inform this draw. Here, the sampling differs from Büschken and Allenby (2016), as ϕ is not integrated out. Instead, the distribution of z is decomposed again into likelihood times prior:

$$p(z_{d,s} \mid \theta, \tau_{d,s,\cdot}, w_{d,s,\cdot}, \phi, \psi, \delta) \propto p(w \mid \phi_{z_{d,s}}) \cdot p(z_{d,s} \mid \theta_d) \tag{B.21}$$

The prior $p(z_{d,s} \mid \theta_d)$ is unchanged from uLDA and still equals θ_d. The likelihood however is now dependent on all topic words within sentence s:

$$\begin{aligned} p(w_{d,s,\cdot} \mid \phi_{z_{d,s}}, \tau_{d,s} = 1) &= \prod_{r \in s} p(w_{d,s,r} \mid \phi_{z_{d,s} \tau_{d,s}}) \\ &= \prod_{r \in s} \left(\phi_{w_{d,s,r}, z_{d,s}}\right)^{\tau_{d,s,r}} \end{aligned} \tag{B.22}$$

The exponent $\tau_{d,s,r}$ leads to only words with $\tau = 1$ being considered in this equation, since $\tau = 0$ reduces the respective probability to the

multiplicative identity 1. If equation B.22 is put into equation B.21, the resulting probability is

$$p(z_{d,s} = t \mid \theta, \tau_{d,s,\cdot}, w_{d,s,\cdot}, \phi) \propto \prod_{r \in s} \left(\phi_{w_{d,s,r}, z_{d,s}}\right)^{\tau_{d,s,r}} \cdot \theta_{d,t} \tag{B.23}$$

which leads to a target distribution of $z_{d,s}$ as:

$$p(z_{d,s} \mid \theta, \tau_{d,s,\cdot}, w_{d,s,\cdot}, \phi) \sim Mult\left(\frac{\prod_{r \in s} \left(\phi_{w_{d,s,r}, z_{d,s}}\right)^{\tau_{d,s,r}} \cdot \theta_d}{\sum_{t \in T} \prod_{r \in s} \left(\phi_{w_{d,s,r}, z_{d,s}}\right)^{\tau_{d,s,r}} \cdot \theta_{d,t}}\right) \tag{B.24}$$

As a consequence, if sentence s contains no topic terms, the probability is only informed by the topic prior θ.

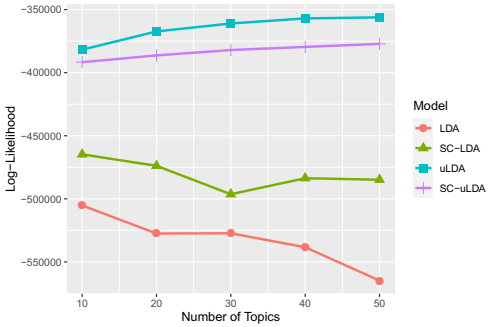
Appendix C

Results

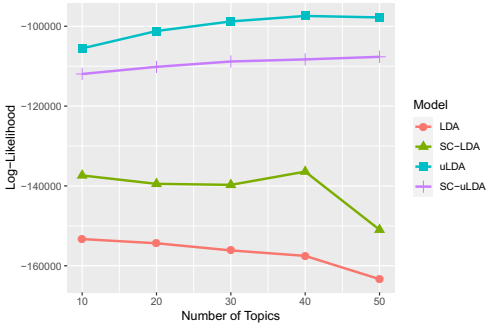
Evaluation results for different topic counts

This section contains a robustness check for the model fit results presented in section 7.2. For all three data sets, each model was estimated five times with different topic counts T ranging from 10 to 50 in increments of 10. Figure C.1 shows the resulting log-likelihood values for the out of sample model fit. The graphics show that the uLDA and SC-uLDA models achieve better results across all data sets and all across all topic numbers. This demonstrates that the findings in the main matter are robust against changes in T.

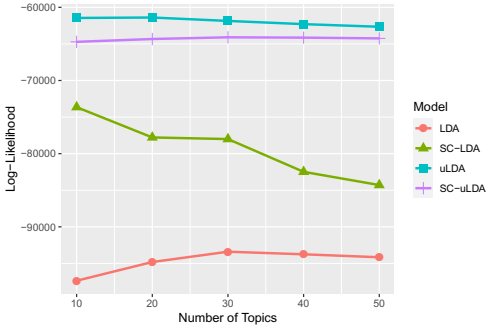
This is mirrored in the results for perplexity. Figure C.2 shows the different perplexity values for different values of T. While the performance of standard LDA and SC-LDA varies across the topic numbers, they are still consistently outperformed by the ubiquitous term extensions. This shows again that the results of this thesis are robust.



(A) Model fit for the *tent* data set.

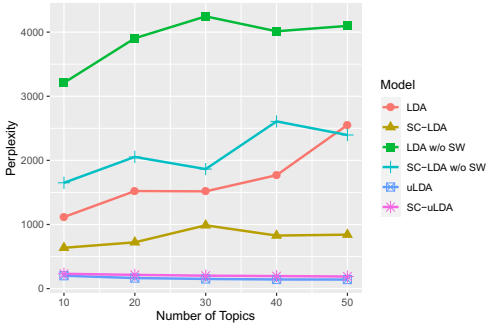


(B) Model fit for the *dogfood* data set.

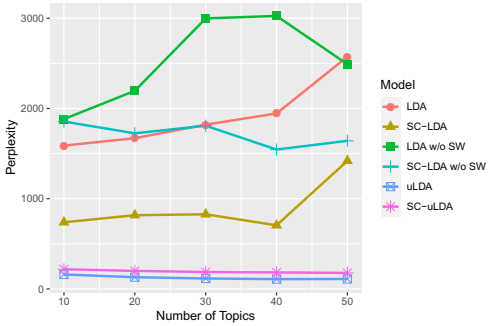


(C) Model fit for the *brexit* data set.

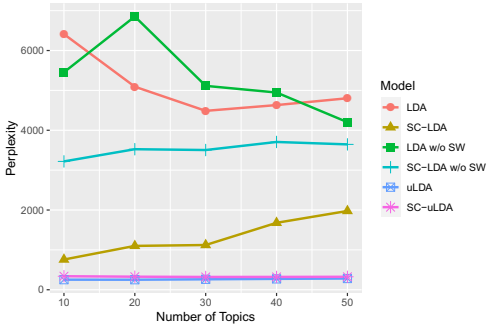
FIGURE C.1: Out of sample log-likelihood across different values for T. Higher values are better.



(A) Model fit for the *tent* data set.



(B) Model fit for the *dogfood* data set.



(C) Model fit for the *brexit* data set.

FIGURE C.2: Out of sample perplexity across different values for T. Lower values are better.

Appendix D

R-Codes

R-Code for Preprocessing

This section contains the R-Code used for pre-processing the data.

```
#### provide a consistent function for preprocessing #####
##
# Input:
# Vector of strings, each string 1 document
#
#

#### standard settings
# removeDuplicates = FALSE
# removeStopwords = FALSE
# customStopwords = character(0)
# contractionHandling = "removegen"
# replaceNumwords = FALSE
# dashReplace = " "
# concatCountwords = TRUE
# concatRanges = FALSE
# concatNegatives = FALSE
# replaceDigits = FALSE
# INCLUDEHITL = FALSE

make_mydocs <- function(documents,
                        removeDuplicates = FALSE,
                        removeStopwords = FALSE,
                        customStopwords = character(0),
                        contractionHandling = "removegen",
                        replaceNumwords = FALSE,
                        dashReplace = "␣",
                        concatCountwords = TRUE,
                        concatRanges = FALSE,
                        concatNegatives = FALSE,
                        replaceDigits = FALSE,
                        tags = NULL){

```

```
34  ### Output: list object with fields:
35  # plaindoc    list of documents with all words
36  #
37  # sentids     list indicating wich word is which sentence
38  # phrids      list indicating which word is which phrase
39  # filters:
40  # stopwords   filter vector list for all stopwords
41  # rarity      vector indicating rarity for each term
42  # shortwords  vector indicating "short" words like "b","k", etc.
43  #
44  # functions
45  # get_docs (
46  #   doc_object:      this object
47  #   split_by:        docs / sent / phr
48  #   keep_stopwords:  yes / no / cov (only keep covariate words)
49  #   rarity:          lower limit (0,1,2,3...)
50  #   keep_shortwords: TRUE/FALSE
51  # )
52
53  ## check if hitl topic tagging was provided
54  if(is.null(tags)){
55    # provide dummy tagging if no tagging present
56    tags <- lapply(1:length(documents), function(x){
57      rep(x, length(documents[[x]]))
58    })
59    include_hitl_coding <- FALSE
60
61  } else {
62    include_hitl_coding <- TRUE
63  }
64
65
66  ## start output list
67  output <- list()
68
69  ### error testing for incompatible settings
70  contractionHandling <- tolower(contractionHandling)
71  if(!(contractionHandling %in% c("pasteall", "removegen", "breakall")
        )){
72    warning("Illegal␣contraction␣handling.␣Will␣use␣'pasteall'.")
73    contractionHandling <- "pasteall"
74  }
75
76  ### remove duplicates of documents
77  if(removeDuplicates){
78    doublettes <- which(duplicated(documents))
79    if(length(doublettes > 0)){
80      documents <- documents[-doublettes]
81      tags <- tags[-doublettes]
82    }
83  }
84
85  N_docs <- length(documents)
86
87
88  ### Define Helpful Constants
```

```
89    Conjunctions <- c("for","and","nor","but","or","yet","so","after","
          although","as","as␣if","as␣long␣as","as␣much␣as","as␣soon␣as","
          as␣though","because","before","even","even␣if","even␣though","
          if","if␣only","if␣when","if␣then","inasmuch","in␣order␣that", "
          just␣as","lest","now","now␣since","now␣that","now␣when","once",
          "provided","provided␣that","rather␣than","since","so␣that","
          supposing","than","that","though","til","unless","until","when"
          ,"whenever","where","whereas", "whichever","where␣if","
          wherever","whether","which","while","who","whoever","why","what
          ","whom","whose")
90    Conjuntions <- sort(Conjunctions)
91    names(Conjunctions) <- Conjunctions
92
93    Punctuation <- c(",",".",";",":","!","?","&", "(",")")
94    names(Punctuation) <- c("comma","fullstop","semicolon","colon","
          exclamation","question","&","para_open","para_close")
95
96    if(length(customStopwords) == 0){
97      myStopwords <- c("a","able","about","across","anyway","anyways","
            after","all","almost","also","am","among","an",
98                       "and","any","are","as","at","be","because","been"
                             ,"but","by","could","dear","did","do","does"
                             ,
99                       "either","else","ever","every","for","from","get"
                             ,"got","had","has","have","he","her","hers",
100                      "him","his","how","however","in","into","is","it"
                             ,"its","just","least","let","like","likely",
101                      "may","me","might","my","neither","nor","of","off
                             ","often","on", "one","only","other","our","
                             own",
102                      "rather","said","say","says","she","should","
                             since","so","some","than","that","the","
                             their","them",
103                      "then","there","these","they","this","to","too","
                             us","wants","was","we","were","what","when",
104                      "where","which","while","who","whom","why","will"
                             ,"with","would","yet","you","your")
105   } else {
106     myStopwords <- customStopwords
107   }
108   ### my short words - nonsensical particles
109   myshortwords = c("b","c","d","e","f","g","h","ii","j","k","l","ll","
          m","n","p","o","r","s","t","u","v","w","x","y","z","xxx","00")
110
111   ### Define first text cleaning function
112   cleanText <- function(x){
113
114     ### All lower case
115     x <- tolower(x)
116
117     ## Handling of Apostrophe and Contractions
118     ## identify mishandled apostrophes (encoding)
119     x <- gsub("â€™", "'", x, fixed=TRUE)
120
121     ## handle mrsmr etcs by removing dots
122     gsub("\\b(mr|mrs|ms|dr|prof)\\.", "\\1", x)
123
124     ## similar stuff, such as abbreviations
```

```
125     gsub("\\b(vs)\\.", "vs", x)
126     #gsub("\\be\\.g\\.", "eg", x)
127     gsub("\\b([a-z])\\.([a-z])\\.([a-z])\\.␣", "\\1\\2\\3␣", x)
128     gsub("\\b([a-z])\\.([a-z])\\.␣", "\\1\\2␣", x)
129
130     ## how to deal with contractions
131     if(contractionHandling == "pasteall"){
132       # just paste everything together by removing '
133       x <- gsub(’’’, "", x)
134       x <- gsub("’", "", x)
135     } else if(contractionHandling == "removegen") {
136       # remove ' if it seems to be possessive
137       x <- gsub("he’s␣", "hes␣", x)
138       x <- gsub("he’s␣", "hes␣", x)
139       x <- gsub("she’s␣", "shes␣", x)
140       x <- gsub("she’s␣", "shes␣", x)
141       x <- gsub("it’s␣", "its␣", x)
142       x <- gsub("it’s␣", "its␣", x)
143       x <- gsub("that’s␣", "thats␣", x)
144       x <- gsub("that’s␣", "thats␣", x)
145       x <- gsub("’s\\b", "", x)
146       x <- gsub("’s\\b", "", x)
147       ## not-contractions
148       ## if they are not meant to be pasted, they have to be resolved
              to "not"
149       if(!concatNegatives){
150         x <- gsub("n’t\\b", "␣not", x)
151         x <- gsub("n’t\\b", "␣not", x)
152         x <- gsub("\\bhav␣not\\b", "have␣not", x)
153         x <- gsub("\\bwo␣not\\b", "will␣not", x)
154         x <- gsub("\\bca␣not\\b", "cannot", x)
155       }
156       ## remove remaining apostrophes
157       x <- gsub(’’’, "", x)
158       x <- gsub("’", "", x)
159     } else if(contractionHandling == "breakall"){
160       ## breakall means replace apostrophes by space
161       if(concatNegatives){
162         warning("Contraction␣Handling␣is␣set␣to␣break␣all␣Contraction,
                ␣this␣is␣not␣compatible␣with␣Contracting␣negations.\n␣
                Negations␣will␣not␣be␣contracted.")
163         concatNegatives <- FALSE
164       }
165       ## break "n't" into not
166       x <- gsub("n’t\\b", "␣not", x)
167       x <- gsub("n’t\\b", "␣not", x)
168       x <- gsub("\\bhav␣not\\b", "have␣not", x)
169       x <- gsub(’’’, "␣", x)
170       x <- gsub("’", "␣", x)
171     }
172     # remove linebreaks
173     x <- gsub("\\n","␣", x, fixed = TRUE)
174     # remove unprintable characters
175     x <- gsub(’[^[:print:]]’, "␣", x)
176     x <- gsub(’[^\\x00-\\x7F]’, "", x, perl = TRUE)
177     ## concat from-to ranges into single expressions
178     if(concatRanges){
179       x <- gsub("([0-9]+)␣*\\-␣*([0-9]+)", "\\1to\\2", x)
```

```
180     }
181     ## Remove all "-" and replace by preset (standard is " ")
182     x <- gsub("-",dashReplace,x, fixed = TRUE)
183     # Remove "wacky" signs
184     x <- gsub("<br>","␣",x)
185     x <- gsub("\\n","␣", x)
186     x <- gsub("[\\/_\\\\©\\*+#%^=><¢íâ®ââ©½⅛£¿¤ï°≫§≪¸¡´²³]", "",x)
187     x <- gsub("[¢íâ®ââ©½⅛£¿¤ï°≫§≪¸¡´²³]", "",x)
188     x <- gsub("~","",x,fixed=TRUE)
189     x <- gsub("|","",x,fixed=TRUE)
190     # Remove Quotes: "  "
191     x <- gsub('\\"','␣', x)
192     ## Handle Dots / Exclamation / Question Marks
193     ## by replacing multiple with space + Sign + space  ( "Test...Test
            ." -> "Test . Test . ")
194     x <- gsub("([\\.!?,;:])+","␣\\1␣",x)
195     x <- gsub("(&)+","␣\\1␣",x)
196     ## Handle Brackets
197     x <- gsub('[\\]\\}\\)]', "␣)␣", x, perl = TRUE)
198     x <- gsub('[\\[\\{\\(]', "␣(␣", x, perl = TRUE)
199     ## Remove all spelled out unicode signs
200     x <- gsub("(\\\\u[0-9a-z])", "", x)
201     ## Handle elongated words
202     x <- gsub("\\bso+\\b", "so", x)
203     ## Handle Numeral words by replacing one - ten with 1 - 10
204     if(replaceNumwords){
205       numwords <- c("one","two","three","four","five","six","seven","
              eight","nine","ten", "eleven","twelve")
206       for(i in 1:length(numwords)){
207         x <- gsub(paste0("\\b(",numwords[i],")\\b"), as.character(i),
                x)
208       }
209     }
210
211     ### handle common num_combinations
212     if(concatCountwords){
213       x <- gsub("\\bstars\\b", "star",x,fixed=TRUE)
214       x <- gsub("([0-9]+)␣(minutes|minute|nights|person|people|stars|
              star|inches)", "\\1\\2", x)
215       x <- gsub("([0-9]+)''", "\\1inches",x)
216     }
217
218     ### Handle Negations by contracting them
219     if(concatNegatives){
220       x <- gsub("do␣not", "dont", x, fixed = TRUE)
221       x <- gsub("did␣not", "didnt", x, fixed = TRUE)
222       x <- gsub("does␣not", "doesnt", x, fixed = TRUE)
223       x <- gsub("has␣not", "hasnt", x, fixed = TRUE)
224       x <- gsub("have␣not", "havnt", x, fixed = TRUE)
225       x <- gsub("can␣not", "cant", x, fixed = TRUE)
226       x <- gsub("cannot", "cant", x, fixed = TRUE)
227       x <- gsub("should␣not", "shouldnt", x, fixed = TRUE)
228     }
229
230     ### remove Dollar and Euro signs
231     x <- gsub("[\\$\\€]+", "␣",x)
232
233     ###### Replace DIGIT?
```

```
234     if(replaceDigits){
235       x <- gsub("[0-9]", "DIGIT", x)
236     }
237     # strip whitespaces
238     x <- gsub("␣+","␣",x)
239     x <- strsplit(x, "␣")
240     x <- lapply(x, function(y) y[y != ""])
241     return(x)
242   }
243
244   ## apply cleantext function
245   docprep <- lapply(documents,cleanText)
246   names(docprep) <- NULL
247   tagprep <- tags
248
249   ### throw out empty reviews  after clean-up
250   non.empty <- lapply(docprep, length) > 0
251   docprep <- docprep[non.empty]
252   tagprep <- tagprep[non.empty]
253
254   ############## next: Build Output List
255   ## first: make some lists
256   #### ID-Vectors for later being able to split the document into
          sentences
257   docid <- list()
258   sentid <- list()
259   phrid <- list()
260   covid <- list()
261
262   #### Filter-Vectors for filtering Stopwords, short words etc
263   filter_stopwords <- list()
264   filter_conjunctions <- list()
265   filter_shortwords <- list()
266
267   ### Punctuation to Split documents
268   fullstops <- Punctuation[c("fullstop","exclamation","question")]
269   all.cov <- append(Conjunctions,Punctuation)
270
271   ## now apply for each document
272   for(i in 1:length(docprep)){
273     ## get temporary doc
274     tmpdoc <- docprep[[i]]
275     tmptag <- rep(tagprep[[i]], sapply(tmpdoc, length))
276     tmpdoc <- unlist(tmpdoc)
277
278     ## indices to seperate by
279     sent.idx <- punct.idx <- allcov.idx <- c(1, rep(0,length(tmpdoc)-1
            ))
280
281     ## mark start of document
282     which.cov <- which(tmpdoc %in% all.cov)
283     names(which.cov) <- tmpdoc[which.cov]
284     tmp.cov <- data.frame(cov = names(which.cov), idx = which.cov)
285
286     sent.idx[tmp.cov$idx[tmp.cov$cov %in% fullstops]] <- 1
287     punct.idx[tmp.cov$idx[tmp.cov$cov %in% Punctuation]] <- 1
288     allcov.idx[tmp.cov$idx[tmp.cov$cov %in% all.cov]] <- 1
289
```

```
290     ## now make running index of which phrase to select
291     sent.split <- cumsum(sent.idx)
292     punct.split <- cumsum(punct.idx)
293     allcov.split <- cumsum(allcov.idx)
294
295     ## Second: remove Punctuation
296     #### remove all punctuation from document, since its unwanted for
            LDA
297     punctfilter <- !(tmpdoc %in% Punctuation)
298
299     ### Remove from DOCS AND TAGS
300     tmpdoc <- tmpdoc[punctfilter]
301     tmptag <- tmptag[punctfilter]
302     ### also remove from split indices
303     sent.split <- sent.split[punctfilter]
304     punct.split <- punct.split[punctfilter]
305     allcov.split <- allcov.split[punctfilter]
306
307     ## save to list
308     docid[[i]] <- rep(1,length(tmpdoc))
309     sentid[[i]] <- sent.split
310     phrid[[i]] <- punct.split
311     covid[[i]] <- allcov.split
312
313     ## Third: get StopWord-Indicator
314     sw_filter <- !(tmpdoc %in% myStopwords)
315     filter_stopwords[[i]] <- sw_filter
316
317     ## Fourth: get conjunction-Indicator
318     conj_filter <- !(tmpdoc %in% Conjunctions)
319     filter_conjunctions[[i]] <- conj_filter
320
321     ## Fifth: get short words indicator
322     shortfilt <- !(tmpdoc %in% myshortwords)
323     filter_shortwords[[i]] <- shortfilt
324
325     ## save updated document (without punctuation)
326     docprep[[i]] <- tmpdoc
327     tagprep[[i]] <- tmptag
328   }
329
330   ## NOW make rarity index
331   ### NEW - rarity by document appearance
332   vocab <- sort(unlist(docprep))
333   rarity_table <- table(vocab) * 0
334   for(i in 1:length(docprep)){
335     docvoc <- unique(unlist(docprep[[i]]))
336     rarity_table[docvoc] <- rarity_table[docvoc] + 1
337   }
338   rarity_index <- list()
339   for(i in 1:length(docprep)){
340     ## get rarity
341     rarity_index[[i]] <- rarity_table[docprep[[i]]]
342   }
343
344   ## save indices to list
345   ID <- list()
346   ID$docs <- docid
```

```
347   ID$sent <- sentid
348   ID$phr <- phrid
349   ID$cov <- covid
350   ## add to output list object
351   output$plaindoc <- docprep
352   output$ID <- ID
353   output$filter_stopwords <- filter_stopwords
354   output$filter_conjunctions <- filter_conjunctions
355   output$filter_shortwords <- filter_shortwords
356   output$rarity_index <- rarity_index
357
358   if(include_hitl_coding){
359     output$coding <- tagprep
360   }
361
362   ###### Filter-Function
363   ## Inputs Object (like the "output"-list)
364   ## Inputs Filter list
365   ##
366   ## outputs filtered object
367   ##
368   output$filter_docs <- function(docobject, fltr){
369     ## for all object fields, filter by fltr
370     for(k in 1:length(docobject$plaindoc)){
371       for(i in 1:length(docobject$ID)){
372         docobject$ID[[i]][[k]] <- docobject$ID[[i]][[k]][fltr[[k]]]
373       }
374       docobject$plaindoc[[k]] <- docobject$plaindoc[[k]][fltr[[k]]]
375       docobject$rarity_index[[k]] <- docobject$rarity_index[[k]][fltr
              [[k]]]
376
377       if(!is.null(docobject$coding)){
378         docobject$coding[[k]] <- docobject$coding[[k]][fltr[[k]]]
379       }
380     }
381     ## now, check if any documents are empty and remove if necessary
382     get_empty <- which(sapply(docobject$plaindoc, length) == 0)
383     if(length(get_empty) > 0){
384       for(i in 1:length(docobject$ID)){
385         docobject$ID[[i]] <- docobject$ID[[i]][-get_empty]
386       }
387       docobject$plaindoc <- docobject$plaindoc[-get_empty]
388       docobject$rarity_index <- docobject$rarity_index[-get_empty]
389       if(!is.null(docobject$coding)){
390         docobject$coding <- docobject$coding[-get_empty]
391       }
392     }
393     return(docobject)
394   }
395
396   #### Function to make document files like "mydoc"
397   ## Inputs
398   # docobj            Document Object, like "output"-list
399   # splitby           How to split up documents
400   #      docs         whole documents (mydoc)
401   #      sent         into sentences (by . ! ?)
402   #      phr          into phrases (by . ! ? , ; : ( ) )
403   #      cov          into phrases (by covariates and punctuation)
```

```
404   # keep_stopwords     whether or not to keep stopwords in the document
405   # rarity             minimum rarity of terms (occurence count)
406   # keep_shortwords    whether or not to keep short terms
407   # get_filter         whether or not to output the filter applied
408   #
409   ## Output
410   # EITHER:
411   #   list of documents
412   # OR:
413   #   list of length 2:
414   #     list of documents
415   #     list of filters used to get to these documents
416   #
417   output$get_docs <- function(docobj,
418                               splitby = "docs",
419                               keep_stopwords = TRUE,
420                               rarity = 2,
421                               keep_shortwords = TRUE,
422                               get_filter = FALSE){
423
424     ## check for different possible inputs for keep_stopwords
425     if(tolower(keep_stopwords) %in% c("y","yes","all","true","t")){
426       keep_stopwords <- TRUE
427     } else if (tolower(keep_stopwords) %in% c("n","none","no","false",
          "f")){
428       keep_stopwords <- FALSE
429     }
430     ## make plain filter
431     use_filter <- lapply(docobj$plaindoc,function(x){
432       rep(TRUE, length(x))
433     })
434
435     ## adjust filter if removing stop words
436     if(keep_stopwords == FALSE){
437       for(i in 1:length(use_filter)){
438         use_filter[[i]] <- docobj$filter_stopwords[[i]]
439       }
440     }
441     ## adjust filter if keeping covariates
442     if(tolower(keep_stopwords) %in% c("cov", "covariates")){
443       for(i in 1:length(use_filter)){
444         use_filter[[i]] <- docobj$filter_stopwords[[i]] | !(docobj$
              filter_conjunctions[[i]])
445       }
446     }
447     ## adjust filter if removing rare terms
448     if(rarity > 0){
449       if(rarity < 1){
450         ## interpret rarity as percentage of documents
451         new_rarity <- (length(docobj$plaindoc) * rarity)
452         rarity <- new_rarity
453       }
454       for(i in 1:length(docobj$plaindoc)){
455         use_filter[[i]] <- use_filter[[i]] & (docobj$rarity_index[[i]]
              > rarity)
456       }
457     }
458     ## adjust filter if removing short terms
```

```
459     if(!keep_shortwords){
460       for(i in 1:length(docobj$plaindoc)){
461         use_filter[[i]] <- use_filter[[i]] & docobj$filter_shortwords
                [[i]]
462       }
463     }
464
465     ## apply filter on object
466     newobj <- docobj$filter_docs(docobj, use_filter)
467     ## if applicable, use topics
468     if(!is.null(newobj$coding)){
469       output_hitl <- TRUE
470     } else {
471       output_hitl <- FALSE
472     }
473
474     ## create docliist according to splitby
475     doclist <- list()
476
477     if(output_hitl) taglist <- newobj$coding
478
479     if(splitby == "docs"){
480       ## if "docs", no splitting needed
481       doclist <- newobj$plaindoc
482     } else {
483       for(i in 1:length(newobj$plaindoc)){
484         tmpdoc <- newobj$plaindoc[[i]]
485         tmpsplit <- newobj$ID[[splitby]][[i]]
486         sentlist <- list()
487         for(j in 1:max(tmpsplit)){
488           sentlist[[j]] <- tmpdoc[tmpsplit == j]
489         }
490         nonzero <- which(sapply(sentlist, length) > 0)
491         doclist[[i]] <- sentlist[nonzero]
492       }
493     }
494
495     ## if prompted, return filter, else return just documents
496     res <- list()
497     res$docs <- doclist
498     if(get_filter){
499       res$filter <- use_filter
500     }
501     if(output_hitl){
502       res$coding <- taglist
503     }
504     if(length(res) == 1){
505       res <- res[[1]]
506     }
507     return(res)
508   }
509
510   meta <- list( removeDuplicates = removeDuplicates,
511                 removeStopwords = removeStopwords,
512                 customStopwords = customStopwords,
513                 contractionHandling = contractionHandling,
514                 replaceNumwords = replaceNumwords,
515                 dashReplace = dashReplace,
```

```
                concatCountwords = concatCountwords,
                concatRanges = concatRanges,
                concatNegatives = concatNegatives,
                replaceDigits = replaceDigits)
  output$meta <- meta
  return(output)
}
```

R-Code Models

R-Code LDA

```
################ PIPELINE ###################
# Step 4: Estimate models on HITL Data
# Substep 1) LDA
#

## requires:
#  docs     list object containing text data
#  nT       integer object containing number of Topics
#  dataset  character naming the dataset in use
#
  # priors:
  # alpha_value
  # beta.phi_value
  # beta.psi_value
  # gamma_value

R = 1500

## allow for overwriting R
if("OVERWRITER" %in% ls()){
  R <- OVERWRITER
}

LISTLENGTH <- 100
LISTLENGTH <- min(LISTLENGTH, R)    # must be smaller than R
ZLLENGTH <- 100                     # list for zdraw is smaller
ZLLENGTH <- min(ZLLENGTH, R)

#### Prepare Data
D <- length(docs)
N_d <- sapply(docs,length)
a <- rep(1:D, N_d)

## setup vocabulary
i2w <- sort(unique(unlist(docs)))
## enable ovveride for holdout
if("OVERRIDEVOCAB" %in% ls()){
  if(class(OVERRIDEVOCAB) == "list"){
    i2w <- OVERRIDEVOCAB$i2w
```

```
  }
}
W <- length(i2w)
w2i <- 1:W
names(w2i) <- i2w

w <- w2i[unlist(docs)]
N <- length(w)

### Priors
alpha = rep(alpha_value,nT)
beta.phi = rep(beta.phi_value,W)
beta.psi = rep(beta.psi_value,W)
gamma = gamma_value


### Starting Values
z = apply(rmultinom(N,1,rep(1/nT,nT)),2,which.max)
### Topic-author word counts
C_ta = matrix(0,nT,D)
C_ta = matrix(table(data.table(cbind(z,a))),nT,D)
# List Format
# TRANSPOSED
L_at = list()
for(i in 1:ncol(C_ta)){
  L_at[[i]] <- C_ta[,i]
}
### Terms-topics counts
C_wt = matrix(0,W,nT)
rows = as.numeric(unlist(labels(table(w)))) # account for all-0 rows
    in C_wt
C_wt[rows,] = table(data.table(cbind(w,z)))
# List Format
L_wt = list()
for(i in 1:nrow(C_wt)){
  L_wt[[i]] <- C_wt[i,]
}
## also store z counts seperately
zcounts = (Reduce("+", L_wt))

## epsilon, sometimes needed to avoid log of zero
eps = 1e-128

### generate starting values
theta = matrix(NA,nT,D)
for(d in 1:D){theta[,d] = MCMCpack::rdirichlet(1,alpha)}

phi = matrix(NA,W,nT)
for(t in 1:nT){phi[,t]=MCMCpack::rdirichlet(1,C_wt[,t]+beta.phi)}

#### Data Infrastructure ####
zdraw <- list()
Theta_draw <- list()
Phi_draw <- list()

# MCMC Sampling
itime = proc.time()[3]

```

```
97  ################### progress bar #####
98  pb <- progress_bar$new(
99    format = "␣Fitting␣LDA:␣[:bar]␣:percent␣eta:␣:eta␣(:tick_rate)",
100   total = R, clear = FALSE, width= 90)
101
102 for (r in 1:R){
103
104   ########################## Draw of z ##################
105   ## collapsed gibbs
106   for(i in 1:N){
107     # Collapsed Gibbs
108     #### List Notation
109     if(L_wt[[w[i]]][z[i]] <= 0 | L_at[[a[i]]][z[i]] <= 0){
110       cat("ERROR:␣Removing␣count␣below␣0")
111     }
112     L_wt[[w[i]]][z[i]] = L_wt[[w[i]]][z[i]] - 1
113     L_at[[a[i]]][z[i]] = L_at[[a[i]]][z[i]] - 1
114     zcounts[z[i]] = zcounts[z[i]] - 1
115     p1 <- (L_wt[[w[i]]]+ beta.phi[w[i]]) / (zcounts + sum(beta.phi))
116     p2 <- (L_at[[a[i]]] + alpha) / sum(L_at[[a[i]]] + alpha)
117     pta = p1 * p2
118     z[i] = which.max(rmultinom(1,1,pta))
119     pta <- NA
120     L_wt[[w[i]]][z[i]] = L_wt[[w[i]]][z[i]] + 1
121     L_at[[a[i]]][z[i]] = L_at[[a[i]]][z[i]] + 1
122     zcounts[z[i]] = zcounts[z[i]] + 1
123   } # End of N Loop
124   ################## Draw of Theta ####################
125   theta <- apply(matrix(unlist(L_at), ncol= length(L_at))
126                  , 2, function(x) MCMCpack::rdirichlet(1,x + alpha))
127   ################## Draw of Phi ####################
128   phi <- apply(matrix(unlist(L_wt), byrow = TRUE, nrow= length(L_wt))
129                , 2, function(x) MCMCpack::rdirichlet(1,x + beta.phi))
130
131   ####### store parameters
132   if(r <= LISTLENGTH){
133     Theta_draw[[r]] <- theta
134     Phi_draw[[r]] <- phi
135   } else {
136     Theta_draw[[1]] <- NULL
137     Phi_draw[[1]] <- NULL
138     Theta_draw[[LISTLENGTH]] <- theta
139     Phi_draw[[LISTLENGTH]] <- phi
140   }
141
142   ## zdraw only save "ZLLENGTH" iterations
143   if(r <= ZLLENGTH){
144     zdraw[[r]] <- z
145   } else {
146     zdraw[[1]] <- NULL
147     zdraw[[ZLLENGTH]] <- z
148   }
149
150   C_wt = matrix(0,W,nT)
151   rows = as.numeric(unlist(labels(table(w)))) # account for all-0 rows
        in C_wt
152   C_wt[rows,] = table(data.table(cbind(w,z)))
153
```

```
  pb$tick()

} # END R Loop
cat("Finished LDA with", nT, "Topcis and", R, "reps.\n")
```

R-Code uLDA

```
################ PIPELINE ###################
# Step 4: Estimate models on HITL Data
# Substep 1) Ubi-LDA
#

## requires:
#  docs     list object containing text data
#  nT       integer object containing number of Topics
#  dataset  character naming the dataset in use
#
  # priors:
  # alpha_value
  # beta.phi_value
  # beta.psi_value
  # gamma_value

R = 1500

## allow for overwriting R
if("OVERWRITER" %in% ls()){
  R <- OVERWRITER
}

LISTLENGTH <- 100
LISTLENGTH <- min(LISTLENGTH, R)    # must be smaller than R
ZLLENGTH <- 100                     # list for zdraw is smaller
ZLLENGTH <- min(ZLLENGTH, R)

#### Prepare Data
D <- length(docs)
N_d <- sapply(docs,length)
a <- rep(1:D, N_d)

## setup vocabulary
i2w <- sort(unique(unlist(docs)))
W <- length(i2w)
w2i <- 1:W
names(w2i) <- i2w
w <- w2i[unlist(docs)]
N <- length(w)

### Priors
alpha = rep(alpha_value,nT)
beta.phi = rep(beta.phi_value,W)
beta.psi = rep(beta.psi_value,W)
gamma = gamma_value
```

```
47
48  ### Starting Values
49  delta = rbeta(1,gamma[1],gamma[2])
50
51  tau = rbinom(N,1,delta)
52  k = sum(tau)
53  z = rep(NA,N)
54  z[tau == 1] = apply(rmultinom(k,1,rep(1/nT,nT)),2,which.max)
55
56  ### Topic-author word counts
57  C_ta = matrix(0,nT,D)
58  C_ta = matrix(table(data.table(cbind(z,a))),nT,D)
59
60  ### Terms-topics counts
61  C_wt = matrix(0,W,nT)
62  rows = as.numeric(unlist(labels(table(w)))) # account for all-0 rows
        in C_wt
63  C_wt[rows,] = table(data.table(cbind(w,z)))
64
65  ### Terms-Ubiquitous-Counts
66  C_w0 = table(data.table(cbind(w,tau)))[,1]
67
68  ## epsilon, sometimes needed to avoid log of zero
69  eps = 1e-128
70
71  ### generate starting values
72  theta = matrix(NA,nT,D)
73  for(i in 1:D){theta[,i] = MCMCpack::rdirichlet(1,C_ta[,i]+alpha)}
74
75  phi = matrix(NA,W,nT)
76  for(t in 1:nT){phi[,t] =  MCMCpack::rdirichlet(1,C_wt[,t]+beta.phi)}
77
78  psi = matrix(NA,W,1)
79  psi[,1] = MCMCpack::rdirichlet(1,C_w0+beta.psi)
80
81  #### Data Infrastructure ####
82  zdraw <- list()
83  taudraw <- list()
84
85  Theta_draw <- list()
86  Phi_draw <- list()
87  Psi_draw <- list()
88
89  naccept_theta = 0
90  deltadraw <- rep(NA,R)
91
92  # MCMC Sampling
93  itime = proc.time()[3]
94  ################### progress bar #####
95  pb <- progress_bar$new(
96    format = "␣Fitting␣Ubi:␣[:bar]␣:percent␣eta:␣:eta␣(:tick_rate)",
97    total = R, clear = FALSE, width= 90)
98
99  for (r in 1:R){
100   ###################### Draw of Tau #################
101   for(i in 1:N){
102     p_t1 = (phi[w[i],]) %*% theta[, a[i]] * delta
103     p_t0 = (psi[w[i]]) * (1-delta)
```

```
104       p_tau = p_t1 / (p_t1 + p_t0)
105       if(is.nan(p_tau)){
106         cat("Error: p_tau is NaN \n p_t1 =",p_t1, "| p_t0 =",p_t0,"...")
107         p_t1 = (phi[w[i],] + eps) %*% theta[, a[i]] * delta
108         p_t0 = (psi[w[i]] + eps) * (1-delta)
109         p_tau = p_t1 / (p_t1 + p_t0)
110         if(!is.nan(p_tau)) {
111           cat("fixed.\n")
112         } else {
113           cat("failed.\n")
114         }
115       }
116
117       tau[i] = rbinom(1,1,p_tau)
118       if(is.na(tau[i])){
119         cat("Error: Tau is NA at i =", i, " | r =",r,"\n", "p_tau =", p
              _tau, "\n")
120       }
121     } # End of N-Loop (tau)
122     ## Update Cw0
123     if(sum(tau) < length(tau)){
124       C_w0 = table(data.table(cbind(w,tau)))[,"0"]
125     } else {
126       C_w0 = rep(0, W)
127     }
128
129     ########################## Draw of z ##################
130     for(i in 1:N){
131       ## full conditional Gibbs
132       if(tau[i] == 1){
133         pz <- phi[w[i],] * theta[,a[i]]
134         if(sum(pz == 0)){
135           pz <- (phi[w[i],] + eps) * (theta[,a[i]] + eps)
136         }
137         z[i] = which.max(rmultinom(1,1,pz))
138       }     else {
139         z[i] = NA
140       }
141     }  # End of N Loop (z)
142     ### Update Counts
143     C_wt = matrix(0,W,nT)
144     rows = as.numeric(unlist(labels(table(w)))) # account for all-0 rows
            in C_wt
145     sel_cols <- as.numeric(unlist(labels(table(z))))
146     C_wt[rows,sel_cols] = table(data.table(cbind(w,z)))
147     C_ta = matrix(0,nT,D)
148     C_ta = matrix(table(data.table(cbind(z,a))),nT,D)
149
150     ################## Draw of Theta ####################
151     theta = matrix(NA,nT,D)
152     for(i in 1:D){theta[,i] = MCMCpack::rdirichlet(1,C_ta[,i]+alpha)}
153
154     ########## Draw of Phi and Psi #############
155     phi = matrix(NA,W,nT)
156     for(t in 1:nT){
157       phi[,t]=MCMCpack::rdirichlet(1,C_wt[,t]+beta.phi)
158     }
159     ### Draw PSI
```

```
  psi = rep(NA,W)
  psi = MCMCpack::rdirichlet(1,C_w0+beta.psi)

  ############ draw delta ############
  k = sum(tau)
  delta = rbeta(1,k+gamma[1],N-k+gamma[2])

  ####### store parameters
  if(r <= LISTLENGTH){
    Theta_draw[[r]] <- theta
    Phi_draw[[r]] <- phi
    Psi_draw[[r]] <- psi
  } else {
    Theta_draw[[1]] <- NULL
    Phi_draw[[1]] <- NULL
    Psi_draw[[1]] <- NULL
    Theta_draw[[LISTLENGTH]] <- theta
    Phi_draw[[LISTLENGTH]] <- phi
    Psi_draw[[LISTLENGTH]] <- psi
  }
  ## zdraw only save "ZLLENGTH" iterations
  if(r <= ZLLENGTH){
    taudraw[[r]] <- tau
    zdraw[[r]] <- z
  } else {
    zdraw[[1]] <- NULL
    taudraw[[1]] <- NULL
    zdraw[[ZLLENGTH]] <- z
    taudraw[[ZLLENGTH]] <- tau
  }
  deltadraw[r] <- delta

  pb$tick()
} # END R Loop
cat("Finished Ubi-LDA with", nT, "Topcis and", R, "reps.\n")
```

R-Code SC-LDA

```
################ PIPELINE ###################
# Step 4: Estimate models on HITL Data
# Substep 2) SC-LDA
#

## requires:
#  docs     list object containing text data
#  nT       integer object containing number of Topics
#  dataset  character naming the dataset in use
#
  # priors:
  # alpha_value
  # beta.phi_value
  # beta.psi_value
  # gamma_value
```

```
16
17 R = 300
18
19 ## allow for overwriting R
20 if("OVERWRITER" %in% ls()){
21   R <- OVERWRITER
22 }
23
24 LISTLENGTH <- 100
25 LISTLENGTH <- min(LISTLENGTH, R)    # must be smaller than R
26 ZLLENGTH <- 100                     # list for zdraw is smaller
27 ZLLENGTH <- min(ZLLENGTH, R)
28
29 #### Prepare Data
30 D <- length(docs)
31 N_d <- sapply(docs,length)
32 a <- rep(1:D, N_d)
33
34 ## setup vocabulary
35 i2w <- sort(unique(unlist(docs)))
36 W <- length(i2w)
37 w2i <- 1:W
38 names(w2i) <- i2w
39 w <- sapply(docs, function(x) sapply(x, function(y) list(w2i[y])))
40 N <- length(w)
41
42 ### Priors
43 alpha = rep(alpha_value,nT)
44 beta.phi = rep(beta.phi_value,W)
45 beta.psi = rep(beta.psi_value,W)
46 gamma = gamma_value
47
48 ### Starting Values
49 N_w = list()
50 N_s = list()
51 A_w = list()
52 A_s = list()
53 for(d in 1:D){
54   N_s[[d]] = length(w[[d]])
55   N_w[[d]] = unlist(lapply(w[[d]],length))
56   A_s[[d]] = rep(d,N_s[[d]])
57   A_w[[d]] = rep(A_s[[d]],N_w[[d]])
58 }
59 z_s = list()
60 z_w = list()
61 ### Generate sentence-topic indicator randomly
62 for(d in 1:D){
63   z_s[[d]] = apply(rmultinom(N_s[[d]],1,rep(1/nT,nT)),2,which.max)
64   z_w[[d]] = rep(z_s[[d]],N_w[[d]])
65 }
66 ### Topic-author word counts
67 C_ta = matrix(0,nT,D)
68 C_ta = matrix(table(data.table(cbind(unlist(z_s),unlist(A_s)))),nT,D)
69 ### Terms-topics counts
70 C_wt = matrix(0,W,nT)
71 rows = as.numeric(unlist(labels(table(unlist(w))))) # account for all-
       0 rows in C_wt
72 C_wt[rows,] = table(data.table(cbind(unlist(w),unlist(z_w))))
```

```
73
74  ## epsilon, sometimes needed to avoid log of zero
75  eps = 1e-128
76
77  ### generate starting values
78  theta = matrix(NA,nT,D)
79  for(d in 1:D){theta[,d] = MCMCpack::rdirichlet(1,alpha)}
80
81  phi = matrix(NA,W,nT)
82  for(t in 1:nT){phi[,t]=MCMCpack::rdirichlet(1,C_wt[,t]+beta.phi)}
83
84
85  #### Data Infrastructure ####
86  zdraw <- list()
87  Theta_draw <- list()
88  Phi_draw <- list()
89
90  # MCMC Sampling
91  itime = proc.time()[3]
92
93  ################### progress bar #####
94  pb <- progress_bar$new(
95    format = "␣Fitting␣SC-LDA:␣[:bar]␣:percent␣eta:␣:eta␣(:tick_rate)",
96    total = R, clear = FALSE, width= 90)
97
98  for (r in 1:R){
99
100   ## draw of z conditional on theta_d
101   ## only in this case is z independent of rating
102   for(d in 1:D){
103
104     for(s in 1:N_s[[d]]){
105
106       # word count in sentence s (ordered!)
107       w.count.s = table(w[[d]][s])
108
109       # labels of unique words in sentence s (ordered!)
110       labels.s = as.numeric(names(w.count.s))
111
112       # number of unique elements of sentence s
113       nu = length(w.count.s)
114
115       C_wt[labels.s,z_s[[d]][s]] = C_wt[labels.s,z_s[[d]][s]] - w.
              count.s
116       C_ta[z_s[[d]][s],A_s[[d]][s]] = C_ta[z_s[[d]][s],A_s[[d]][s]] -
              1
117
118       # initialize matrices A and B with standard values
119       BigA = matrix(C_wt[labels.s,] + beta.phi[labels.s], nu, nT)
120       BigB = matrix(rep(apply(C_wt,2,sum) + sum(beta.phi),nu),nu,nT,
              byrow = TRUE)
121       BigB = log(BigB)
122       BigA = log(BigA)
123
124       ### compute A and B by row
125       for(j in 1:nu){
126         w_u = labels.s[j]
127         # compute count of occurrences of each unique word
```

```
128           count = w.count.s[j]
129           # generating row of matrix A given unique word in sentence s
130           # count == 1 means no change
131           if(count > 1){
132             aa = exp(BigA[j,])
133             reduce_count = c(0:(count-1))
134             aa = rbind(aa,matrix(1,max(reduce_count),nT))
135             aa[,z_s[[d]][s]] = rep(aa[1,z_s[[d]][s]],length(reduce_count
                    ))+reduce_count
136             BigA[j,] = apply(log(aa),2,sum)
137           }
138           ### compute B
139           if(count > 1){
140             bb = exp(BigB[j,])
141             bb = rbind(bb,matrix(1,max(reduce_count),nT))
142             bb[,z_s[[d]][s]] = rep(bb[1,z_s[[d]][s]],length(reduce_count
                    ))+reduce_count
143             BigB[j,] = apply(log(bb),2,sum)
144           }
145         }
146         BigA = apply((BigA),2,sum) # taking logs throughout for
                numerical stability
147         BigB = apply((BigB),2,sum)
148         BigC = C_ta[,d] + alpha
149         BigC = BigC / sum(BigC)
150
151         pt = exp(BigA-BigB+log(BigC))
152         if(identical(rep(0,nT),pt)){            ## this happens when
                values are too small
153                                                 ## due to the large
                                                        vocabulary
154           pt = exp((BigA-BigB+log(BigC))/4)
155           bigsum <- BigA-BigB+log(BigC)
156           if(min(bigsum) < -700)               ## try shift to the right
                by adding stuff to bigsum (so exp(bisgum) > 0)
157           {
158             bsdiff <- (-700 -min(bigsum))
159             if(max(bigsum) + bsdiff > 700){   ## if thats not possible
                    try to squeeze bigsum in [-700,700]
160               # squeeze
161               bsnorm <- bigsum - mean(bigsum)
162               bigsum <- bsnorm /(max(abs(bsnorm))/700)
163             } else {
164               bigsum <- bigsum + (-700 - min(bigsum))
165             }
166             pt = exp(bigsum)
167             if(!identical(rep(0,nT),pt) & max(pt)!=Inf){
168               #cat("...fixed\n")
169             } else {
170               cat("pt error: doc", d,"sentence",s)
171               pt = exp((BigA-BigB+log(BigC))/4)
172               cat("...failed to fix\n")
173             }
174           } else {
175             cat("pt error: doc", d,"sentence",s)
176             pt = exp((BigA-BigB+log(BigC))/4)
177             cat("...cause unknown\n")
178           }
```

```
179       }
180       # draw z
181       z_s[[d]][s] = which.max(rmultinom(1,1,pt)) # vectorizing the pta
              matrix is equivalent
182       rm(pt, BigA, BigB, BigC)
183       ## update C_wt
184       C_wt[labels.s,z_s[[d]][s]] = C_wt[labels.s,z_s[[d]][s]] + w.
              count.s
185       C_ta[z_s[[d]][s],A_s[[d]][s]] = C_ta[z_s[[d]][s],A_s[[d]][s]] +
              1
186     } # s loop
187     # expanding the new topic
188     z_w[[d]] = rep(z_s[[d]],N_w[[d]])
189   } # d loop
190
191   ################### Draw of Theta ######################
192   theta <- apply(C_ta, 2, function(x) MCMCpack::rdirichlet(1,x + alpha
          ))
193
194   ################### Draw of Phi ######################
195   phi <- apply(C_wt, 2, function(x) MCMCpack::rdirichlet(1,x + beta.
          phi))
196
197   ####### store parameters
198   if(r <= LISTLENGTH){
199     Theta_draw[[r]] <- theta
200     Phi_draw[[r]] <- phi
201   } else {
202     Theta_draw[[1]] <- NULL
203     Phi_draw[[1]] <- NULL
204     Theta_draw[[LISTLENGTH]] <- theta
205     Phi_draw[[LISTLENGTH]] <- phi
206   }
207
208   ## zdraw only save "ZLLENGTH" iterations
209   if(r <= ZLLENGTH){
210     zdraw[[r]] <- z_w
211   } else {
212     zdraw[[1]] <- NULL
213     zdraw[[ZLLENGTH]] <- z_w
214   }
215   pb$tick()
216 } # END R Loop
217 cat("Finished␣SC-LDA␣with", nT, "Topcis␣and", R, "reps.\n")
```

R-Code SC-uLDA

```
1 ################ PIPELINE ###################
2 # Step 4: Estimate models on HITL Data
3 # Substep 4) SC-UBI
4 #
5
6 ## requires:
```

```
#  docs     list object containing text data
#  nT       integer object containing number of Topics
#  dataset  character naming the dataset in use
#
  # priors:
  # alpha_value
  # beta.phi_value
  # beta.psi_value
  # gamma_value

R = 2000

## allow for overwriting R
if("OVERWRITER" %in% ls()){
  R <- OVERWRITER
}

LISTLENGTH <- 100
LISTLENGTH <- min(LISTLENGTH, R)    # must be smaller than R
ZLLENGTH <- 100                     # list for zdraw is smaller
ZLLENGTH <- min(ZLLENGTH, R)

#### Prepare Data
D <- length(docs)
N_d <- sapply(docs,length)
a <- rep(1:D, N_d)

## setup vocabulary
i2w <- sort(unique(unlist(docs)))
W <- length(i2w)
w2i <- 1:W
names(w2i) <- i2w
w <- sapply(docs, function(x) sapply(x, function(y) list(w2i[y])))
N <- length(w)

### Priors
alpha = rep(alpha_value,nT)
beta.phi = rep(beta.phi_value,W)
beta.psi = rep(beta.psi_value,W)
gamma = gamma_value

### Starting Values
delta = 0.5 #rbeta(1,gamma[1],gamma[2])

N_w = list()
N_s = list()
A_w = list()
A_s = list()
for(d in 1:D){
  if(class(w[[d]]) == "list"){
    N_s[[d]] = length(w[[d]])
    N_w[[d]] = unlist(lapply(w[[d]],length))
    A_s[[d]] = rep(d,N_s[[d]])
    A_w[[d]] = rep(A_s[[d]],N_w[[d]])
  } else if(class(w[[d]]) %in% c("matrix", "integer")){
    N_s[[d]] = 1
    N_w[[d]] = length(w[[d]])
    A_s[[d]] = rep(d,N_s[[d]])
```

```
    A_w[[d]] = rep(A_s[[d]],N_w[[d]])
  }

}
z_s = list()
z_w = list()
tau = list()

### Generate sentence-topic indicator randomly
for(d in 1:D){
  tau[[d]] <- list()
  S_w <- rep(1:N_s[[d]],N_w[[d]])
  z_s[[d]] = apply(rmultinom(N_s[[d]],1,rep(1/nT,nT)),2,which.max)
  z_w[[d]] = rep(z_s[[d]],N_w[[d]])
  tau_d = rbinom(sum(N_w[[d]]), 1, delta)
  z_w[[d]][tau_d == 0] = NA   # set topic NA for nontopics
  if(sum(tau_d) == 0){
    s_na <- rep(TRUE, N_s[[d]])
  } else {
    s_na <- table(tau_d,S_w)["1",] == 0
  }
  for(s in 1:N_s[[d]]){
    tau[[d]][[s]] <- tau_d[S_w == s]
  }
}
### Topic-author word counts
C_ta = matrix(0,nT,D)
C_ta = matrix(table(data.table(cbind(unlist(z_s),unlist(A_s)))),nT,D)
### Terms-topics counts
C_wt = matrix(0,W,nT)
rows = as.numeric(unlist(labels(table(unlist(w))))) # account for all-
    0 rows in C_wt
sel_cols = as.numeric(unlist(labels(table(unlist(z_w)))))
wtmp <- unlist(w)[unlist(tau)]
C_wt[rows,sel_cols] = table(data.table(cbind(unlist(w),unlist(z_w))))

## Term-Nontopic
C_w0 = matrix(0,W,1)
rows = as.numeric(unlist(labels(table(unlist(w)[unlist(tau) == 0]))))
C_w0[rows,] = table(data.table(unlist(w)[unlist(tau)==0]))

## epsilon, sometimes needed to avoid log of zero
eps = 1e-128

### generate starting values
theta = matrix(NA,nT,D)
for(d in 1:D){theta[,d] = MCMCpack::rdirichlet(1,C_ta[,d] + alpha)}

phi = matrix(NA,W,nT)
for(t in 1:nT){phi[,t] =  MCMCpack::rdirichlet(1,C_wt[,t]+beta.phi)}
# for(t in 1:nT){phi[,t] =  MCMCpack::rdirichlet(1,beta.phi)}

psi = matrix(NA,W,1)
psi[,1] = MCMCpack::rdirichlet(1,C_w0+beta.psi)

#### Data Infrastructure ####
zdraw <- list()
taudraw <- list()
```

```
122
123 Theta_draw <- list()
124 Phi_draw <- list()
125 Psi_draw <- list()
126
127 naccept_theta = 0
128 deltadraw <- rep(NA,R)
129
130 # MCMC Sampling
131 itime = proc.time()[3]
132 ################### progress bar #####
133 pb <- progress_bar$new(
134   format = "␣Fitting␣SC-Ubi:␣[:bar]␣:percent␣eta:␣:eta␣(:tick_rate)",
135   total = R, clear = FALSE, width= 90)
136
137 for (r in 1:R){
138   ###################### Draw of Tau #################
139   ## draw of tau (integrated over z to avoid "empty" sentences)
140   for(i in 1:D){
141     for(s in 1:N_s[[i]]){
142       w.s = w[[i]][[s]]
143       for(ww in 1:length(w.s)){
144         # draw tau sentence-wise
145         p.tau.1 <- phi[w.s[ww],z_s[[i]][[s]]]  * delta
146         p.tau.0 <- psi[w.s[ww],] * (1-delta)
147         p.tau <- p.tau.1 / (p.tau.1 + p.tau.0)
148         tau[[i]][[s]][ww] <- rbinom(1,1,p.tau)
149         if(any(is.na(tau[[i]][[s]]))) stop("tau␣NA")
150       }
151     } # sentence loop
152   } # document loop
153   ## draw of z conditional on theta_d
154   for(i in 1:D){
155     for(s in 1:length(w[[i]])){
156       if(length(w[[i]][[s]][tau[[i]][[s]]==1])==1){  # sentence has 1
              topic word
157         p.z = phi[w[[i]][[s]][tau[[i]][[s]]==1],] * theta[,i]
158         z_s[[i]][[s]] = which.max(rmultinom(1,1,p.z))
159       }
160       if(length(w[[i]][[s]][tau[[i]][[s]]==1])>1){   # sentence has
              several topic words
161         p.z = apply(phi[w[[i]][[s]][tau[[i]][[s]]==1],],2,prod)*theta
              [,i]
162         if(sum(p.z) == 0){
163           p.z = (apply(phi[w[[i]][[s]][tau[[i]][[s]]==1],],2,prod) +
                eps)*(theta[,i] + eps)
164         }
165         z_s[[i]][[s]] = which.max(rmultinom(1,1,p.z))
166       }
167       if(length(w[[i]][[s]][tau[[i]][[s]]==1])==0){  # sentence has no
            topic words
168         ### draw from prior
169         z_s[[i]][[s]] = which.max(rmultinom(1,1,theta[,i]))
170       }
171     } # sentence loop
172   } # document loop
173
174   ## update delta (prior to tau)
```

```
  taucounts = table(unlist(tau))
  tau1 <- taucounts["1"]
  tau0 <- taucounts["0"]
  tau1 <- ifelse(is.na(tau1),0, tau1)
  tau0 <- ifelse(is.na(tau0),0, tau0)

  delta = rbeta(1,tau1+gamma[1],tau0+gamma[2])

  # # expand topic assigment to word level, no topic for non-topic
      words ("NA")
  z_w <- lapply(1:N, function(i){
    zw <- rep(z_s[[i]], N_w[[i]])
    zw[unlist(tau[[i]]) == 0] <- NA
    return(zw)
  })

  ## Update Count Matrices
  ### Topic-author word counts
  C_ta = matrix(0,nT,D)
  C_ta = matrix(table(data.table(cbind(unlist(z_s),unlist(A_s)))),nT,D
      )

  ### Terms-topics counts
  C_wt = matrix(0,W,nT)
  rows = as.numeric(unlist(labels(table(unlist(w))))) # account for
      all-0 rows in C_wt
  sel_cols <- as.numeric(unlist(labels(table(unlist(z_w)))))
  C_wt[rows,sel_cols] = table(data.table(cbind(unlist(w),unlist(z_w)))
      )
  ## Term-Nontopic
  C_w0 = matrix(0,W,1)
  rows = as.numeric(unlist(labels(table(unlist(w)[unlist(tau) == 0]))))
      )
  C_w0[rows,] = table(data.table(unlist(w)[unlist(tau)==0]))

  theta = matrix(NA,nT,D)
  for(d in 1:D){theta[,d] = MCMCpack::rdirichlet(1,C_ta[,d] + alpha)}

  phi = matrix(NA,W,nT)
  for(t in 1:nT){phi[,t] =  MCMCpack::rdirichlet(1,C_wt[,t]+beta.phi)}

  psi = matrix(NA,W,1)
  psi[,1] = MCMCpack::rdirichlet(1,C_w0+beta.psi)

  ####### store parameters
  if(r <= LISTLENGTH){
    Theta_draw[[r]] <- theta
    Phi_draw[[r]] <- phi
    Psi_draw[[r]] <- psi
  } else {
    Theta_draw[[1]] <- NULL
    Phi_draw[[1]] <- NULL
    Psi_draw[[1]] <- NULL
    Theta_draw[[LISTLENGTH]] <- theta
    Phi_draw[[LISTLENGTH]] <- phi
    Psi_draw[[LISTLENGTH]] <- psi
  }

```

```
228   ## zdraw only save "ZLLENGTH" iterations
229   if(r <= ZLLENGTH){
230     taudraw[[r]] <- tau
231     zdraw[[r]] <- z_w
232   } else {
233     zdraw[[1]] <- NULL
234     taudraw[[1]] <- NULL
235     zdraw[[ZLLENGTH]] <- z_w
236     taudraw[[ZLLENGTH]] <- tau
237   }
238   deltadraw[r] <- delta
239
240   pb$tick()
241 } # END R Loop
242 cat("Finished Ubi-SCLDA with", nT, "Topcis and", R, "reps.\n")
```

References

Arun, R. et al. (2010). "On finding the natural number of topics with Latent Dirichlet Allocation: Some observations". In: *Advances in Knowledge Discovery and Data Mining. PAKDD 2010. Lecture Notes in Computer Science*. Ed. by Mohammed J. Zaki et al. Vol. 6118. PART 1. Berlin, Heidelberg: Springer, pp. 391–402. URL: https://link.springer.com/chapter/10.1007/978-3-642-13657-3_43.

Blei, David M., Thomas L. Griffiths, et al. (2004). "Hierarchical Topic Models and the Nested Chinese Restaurant Process". In: *Advances in neural information processing systems*, pp. 17–28.

Blei, David M. and John D. Lafferty (2009). "Topic Models". In: *Text Mining: Classification, Clustering, and Applications*. Ed. by Ashok N. Srivastava and Mehran Sahami. CRC Press, pp. 71–94.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan (2003). "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan, pp. 993–1022.

Büschken, Joachim and Greg M. Allenby (Nov. 2016). "Sentence-based text analysis for customer reviews". In: *Marketing Science* 35.6, pp. 953–975. ISSN: 1526548X. DOI: 10.1287/mksc.2016.0993.

Cao, Donglin et al. (Aug. 2016). "Visual sentiment topic model based microblog image sentiment analysis". In: *Multimedia Tools and Applications* 75.15, pp. 8955–8968. ISSN: 15737721. DOI: 10.1007/s11042-014-2337-z.

Coleman, Meri and T. L. Liau (Apr. 1975). "A computer readability formula designed for machine scoring". In: *Journal of Applied Psychology* 60.2, pp. 283–284. ISSN: 00219010. DOI: 10.1037/h0076540.

Davies, Mark (2008). *The Corpus of Contemporary American English*.

Deerwester, Scott et al. (Sept. 1990). "Indexing by latent semantic analysis". In: *Journal of the American Society for Information Science* 41.6, pp. 391–407.

Do, Trinh-Minh-Tri and Daniel Gatica-Perez (2010). "By their apps you shall understand them". In: *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia - MUM '10*. New York, New York, USA: Association for Computing Machinery (ACM), pp. 1–10. DOI: 10.1145/1899475.1899502.

Dolamic, Ljiljana and Jacques Savoy (2010). "When Stopword Lists Make the Difference". In: *Journal of the American Society for Information Science and Technology* 61.1, pp. 200–203.

Fox, Christopher (Jan. 1989). "A Stop List for General Text". In: *ACM SIGIR Forum* 24.1-2, pp. 19–21. ISSN: 01635840. DOI: 10.1145/378881.378888.

Gilks, Walter R., Sylvia Richardson, and David J. Spiegelhalter (1995). *Markov Chain Monte Carlo in Practice*. Chapman and Hall.

Gormley, Isobel Claire and Thomas Brendan Murphy (2014). "Mixed Membership Models for Rank Data: Investigating Structure in Irish Voting Data". In: *Handbook of Mixed Membership Models and Their Applications*. Ed. by Edoardo M. Airoldi et al. CRC press, pp. 441–460.

Greene, Derek, Derek O'Callaghan, and Pádraig Cunningham (2014). "How many topics? Stability analysis for topic models". In: *ECML PKDD 2014: Machine Learning and Knowledge Discovery in Databases*. Vol. 8724 LNAI. PART 1. Springer Verlag, pp. 498–513.

Griffiths, Thomas L. and Mark Steyvers (2002). "A probabilistic approach to semantic representation". In: *Proceedings of the Twenty-Fourth Annual Conference of Cognitive Science Society*. Vol. 24. 24.

Griffiths, Thomas L., Mark Steyvers, et al. (2004). "Integrating Topics and Syntax". In: *Advances in Neural Information Processing Systems* 17.

Grimmer, Justin (2010). "A Bayesian hierarchical topic model for political texts: Measuring expressed agendas in senate press releases". In: *Political Analysis* 18.1, pp. 1–35. ISSN: 14764989. DOI: 10.1093/pan/mpp034.

Gruber, Amit, Michal Rosen-Zvi, and Yair Weiss (Mar. 2007). "Hidden Topic Markov Models". In: *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 163–170. URL: http://proceedings.mlr.press/v2/gruber07a.html.

Heinrich, Gregor (Sept. 2009). *Parameter estimation for text analysis*. Tech. rep. Darmstadt: Fraunhofer IGD.

Hofmann, Thomas (1999). "Probabilistic Latent Semantic Analysis". In: *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 289–296. URL: http://arxiv.org/abs/1301.6705.

Ishigaki, Tsukasa et al. (Feb. 2015). "Topic Modeling of Market Responses for Large-Scale Transaction Data". In: *DSSR Discussion Papers*. URL: https://ideas.repec.org/p/toh/dssraa/35.html.

Jacobs, Bruno J.D., Bas Donkers, and Dennis Fok (May 2016). "Model-based purchase predictions for large assortments". In: *Marketing Science* 35.3, pp. 389–404. ISSN: 1526548X. DOI: 10.1287/mksc.2016.0985.

Kuhn, Harold W. (Mar. 1955). "The Hungarian method for the assignment problem". In: *Naval Research Logistics Quarterly* 2.1-2, pp. 83–97. ISSN: 00281441. DOI: 10.1002/nav.3800020109.

Mei, Qiaozhu, Xuehua Shen, and Chengxiang Zhai (2007). "Automatic labeling of multinomial topic models". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 490–499. DOI: 10.1145/1281192.1281246.

Moody, Christopher E. (May 2016). "Mixing Dirichlet Topic Models and Word Embeddings to Make lda2vec". In: *arXiv preprint*. URL: http://arxiv.org/abs/1605.02019.

Munkres, James (Mar. 1957). "Algorithms for the Assignment and Transportation Problems". In: *Journal of the Society for Industrial and Applied Mathematics* 5.1, pp. 32–38. ISSN: 0368-4245. DOI: 10.1137/0105003.

Nigam, Kamal et al. (2000). "Text classification from labeled and unlabeled documents using EM". In: *Machine Learning* 39.2, pp. 103–134. ISSN: 08856125. DOI: 10.1023/a:1007692713085.

Packard, Grant, Sarah G. Moore, and Brent McFerran (2018). "(I'm) Happy to Help (You): The Impact of Personal Pronoun Use in Customer–Firm Interactions". In: *Journal of Marketing Research* 55.4, pp. 541–555. ISSN: 0022-2437. DOI: 10.1509/jmr.16.0118.

Pennebaker, James W. (Sept. 2011). "The secret life of pronouns". In: *New Scientist* 211.2828, pp. 42–45. ISSN: 02624079. DOI: 10.1016/S0262-4079(11)62167-2.

Ramage, Daniel, Susan Dumais, and Dan Liebling (2010). "Characterizing Microblogs with Topic Models". In: *4th International AAAI Conference on Weblogs and Social Media*.

Reck, Ronald P. and Ruth A. Reck (2007). "Generating and Rendering Readability Scores for Project Gutenberg Texts". In: *Proceedings of the Corpus Linguistics Conference*. Ed. by Matthew Davies et al.

Rosen-Zvi, Michal et al. (Jan. 2010). "Learning author-topic models from text corpora". In: *ACM Transactions on Information Systems* 28.1, pp. 1–38. ISSN: 10468188. DOI: 10.1145/1658377.1658381.

Schofield, Alexandra, Måns Magnusson, and David Mimno (2017). "Pulling Out the Stops: Rethinking Stopword Removal for Topic

Models". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Vol. 2. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 432–436. ISBN: 9781510838604. DOI: 10.18653/v1/E17-2069.

Shafiei, M. Mahdi and Evangelos E. Muios (2006). "Latent dirichlet co-clustering". In: *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 542–551. ISBN: 0769527019. DOI: 10.1109/ICDM.2006.94.

Shringarpure, Suyash and Eric P. Xing (2014). "Population Stratification with Mixed Membership Models". In: *Handbook of mixed membership models and their applications*. Ed. by Edoardo M. Airoldi et al. CRC press.

Steyvers, M. and T. Griffiths (2010). "Probabilistic Topic Models". In: *Latent Semantic Analysis: A Road To Meaning*. Ed. by T. Landauer et al. Vol. 3. 3. Hillsdale, NJ: Laurence Erlbaum, pp. 993–1022.

Teh, Yee Whye et al. (2006). "Hierarchical Dirichlet Processes". In: *Journal of the American Statistical Association* 101.476, pp. 1566–1581.

UCI (1999). *Reuters-21578 Dataset*. URL: http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html.

Van Valen, Leigh (Jan. 2005). "The statistics of variation". In: *Variation*. Elsevier Inc., pp. 29–47. ISBN: 9780120887774. DOI: 10.1016/B978-012088777-4/50005-3.

Wallach, Hanna M. (2006). "Topic modeling: Beyond bag-of-words". In: *ICML '06: Proceedings of the 23rd international conference on Machine learning*. Vol. 148. New York, New York, USA: ACM Press, pp. 977–984. ISBN: 1595933832. DOI: 10.1145/1143844.1143967.

Wallach, Hanna M., David Mimno, and Andrew Mccallum (2009). "Rethinking LDA: Why Priors Matter". In: *Advances in Neural Information Processing Systems*. Vol. 22, pp. 1973–1981.

Wallach, Hanna M., Iain Murray, et al. (2009). "Evaluation methods for topic models". In: *ACM International Conference Proceeding Series*. Vol. 382. New York, New York, USA: ACM Press, pp. 1–8. ISBN: 9781605585161. DOI: 10.1145/1553374.1553515.

Yang, Zaihan et al. (Aug. 2015). "Parametric and non-parametric user-aware sentiment topic models". In: *SIGIR 2015 - Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, New York, USA: Association for Computing Machinery, Inc, pp. 413–422. ISBN: 9781450336215. DOI: 10.1145/2766462.2767758.

Zhao, Weizhong et al. (Sept. 2015). "A heuristic approach to determine an appropriate number of topics in topic modeling". In: *BMC Bioinformatics* 16.13, pp. 1–10. ISSN: 14712105. DOI: 10.1186/1471-2105-16-S13-S8.