dortmund university

fakultät statistik

Reliability evaluation and an update algorithm for the latent Dirichlet allocation

Dissertation

by

Jonas Rieger

born in Wermelskirchen

in partial fulfillment of
the requirements for the degree of
Doktor der Naturwissenschaften (Dr. rer. nat.)

Submitted: Dortmund, July 2022
Primary referee: Prof. Dr. Carsten Jentsch
Secondary referee: Prof. Dr. Jörg Rahnenführer

Day of oral examination: September 29, 2022

Jonas Rieger

Reliability evaluation and an update algorithm for the latent Dirichlet allocation

Dissertation
in partial fulfillment of
the requirements for the degree of
Doktor der Naturwissenschaften (Dr. rer. nat.)

Submitted to the
Department of Statistics of the
TU Dortmund University

Dortmund, July 2022

Primary referee: Prof. Dr. Carsten Jentsch
Secondary referee: Prof. Dr. Jörg Rahnenführer

Day of oral examination: September 29, 2022

DOI: 10.17877/DE290R-22949.

I warmly thank Carsten Jentsch and Jörg Rahnenführer for supervising my thesis, giving me the guidance I needed and the freedom to follow my own ideas in an appropriate balance.

Ich danke meinen Eltern für alle Anstrengungen, die ihr jemals für mich unternommen habt, für alle Möglichkeiten, die ihr mir geschenkt habt und für die Art und Weise zu leben, die ihr mir gezeigt habt.

Danke an meine Frau, dass du mich jederzeit unterstützt.

Contents

Abstract

Modeling text data is becoming increasingly popular. Topic models and in particular the latent Dirichlet allocation (LDA) represent a large field in text data analysis. In this context, the problem exists that running LDA repeatedly on the same data yields different results. This lack of reliability can be improved by repeated modeling and a reasonable choice of a representative. Further, updating existing LDA models with new data is another common challenge. Many dynamic models, when adding new data, also update parameters of past time points, thus do not ensure the temporal consistency of the results.

In this cumulative dissertation, I summarize in particular my methodological papers from the two areas of improving the reliability of LDA results and updating LDA results in a temporally consistent manner for use in monitoring scenarios. For this purpose, I first introduce the state of research for each of the two areas. After explaining the idea of the corresponding method, I give examples of applications in which the method has already been used and explain the implementation as an R package. Finally, for both fields I provide an outlook on potential further research.

Zusammenfassung

Die Modellierung von Textdaten erfährt wachsende Popularität. Einen großen Bereich in der Textdatenanalyse bilden Topic Modelle und dabei im Speziellen das Modell latent Dirichlet allocation (LDA). Dabei existiert die Problematik, dass sich bei einer wiederholten Ausführung der LDA auf denselben Daten verschiedene Resultate ergeben. Dieser Mangel an Reliabilität lässt sich durch eine wiederholte Modellierung und eine sinnvolle Wahl eines Repräsentanten verbessern. Eine weitere Herausforderung stellt das Aktualisieren von bestehenden LDA-Modellen anhand neuer Daten dar. Viele dynamische Modelle aktualisieren im Falle einer Hinzunahme neuer Daten auch Parameter vergangener Zeitpunkte und verletzen damit die zeitliche Konsistenz der Ergebnisse.

In dieser kumulativen Dissertation fasse ich insbesondere meine methodischen Paper aus den beiden Themenbereichen der Verbesserung der Reliabilität von LDA-Ergebnissen und der zeitlich konsistenten Aktualisierung von LDA-Ergebnissen zur Nutzung in Monitoring-Szenarien zusammen. Dafür stelle ich zunächst jeweils den Forschungsstand dar. Nach einer Erläuterung der Idee der Methode, werden jeweils Beispiele gegeben, in denen die Methode bereits Anwendung fand und die Implementierung als R Paket erläutert. Zuletzt gebe ich für beide Themenbereiche einen Ausblick auf mögliche weitere Forschung.

Abbreviations

AJC	average Jaccard coefficient
ATM	author-topic model
BERT	bidirectional encoder representations from transformers
BLEU	bilingual evaluation understudy (score)
cDTM	continuous time dynamic topic model
CGS	collapsed Gibbs sampler
CVB	collapsed variational Bayes
CTM	correlated topic model
dDTM	discrete time dynamic topic model
DTM	document-term matrix
ecdf	empirical cumulative distribution function
EM	expectation-maximization (algorithm)
EPU	economic policy uncertainty (index)
ETM	embedded topic model
EU	European Union
GDP	gross domestic product
GLDA	granulated latent Dirichlet allocation
IPI	inflation perception indicator
JC	Jaccard coefficient
JSD	Jensen-Shannon divergence
KLD	Kullback-Leibler divergence
LDA	latent Dirichlet allocation
LSI	latent semantic indexing
MAP	maximum a posteriori (estimation)
MCMC	Markov chain Monte Carlo (method)
ML	maximum likelihood (estimation)
NLP	natural language processing
NPMI	normalized pointwise mutual information
NTM	neural topic model
PLSA	probabilistic latent semantic analysis
PLSI	probabilistic latent semantic indexing

PMI	pointwise mutual information
RBO	rank-biased overlap
S-CLOP	similarity of multiple sets by clustering with local pruning
STM	structural topic model
SVD	singular value decomposition
tf-idf	term frequency-inverse document frequency
TM-LDA	temporal LDA
TOT	topics over time
UPI	uncertainty perception indicator
VB	variational Bayes

1 Introduction

In January 2009, Google's chief economist Hal Varian expressed in *The McKinsey Quarterly* the famous quote, "I keep saying the sexy job in the next ten years will be statisticians. People think I'm joking, but who would've guessed that computer engineers would've been the sexy job of the 1990s?". While in fact the designation Data Scientist turned out to be the sexy one in the following ten years, it can be argued that he actually was right about the content of his statement. One year after this quote, in 2010, there was already a total volume of 2 zettabytes of data created, captured, copied, and consumed globally, according to Statista. They report huge growth over ten years with a volume of already 64.2 zettabytes for 2020, which they estimate to nearly triple again until 2025.[1] While these numbers do not provide any direct indication of a possible increase in the relevance of the Data Scientist job, it is certainly clear that these data must be, or have been, given meaning and interpretation through analysis and processing. After all, it is evident that the analysis of existing data in turn generates new amounts of data.

Thereby the attention for what are actual (new) findings, what is only another form of representation of already known facts, and what are possibly no newly discovered systematics, but coincidental dependencies, which are wrongly interpreted as nature-given relations due to our in the last years newly acquired - and partly not mastered - abilities and resources; the sensitivity for these relations must not be lost. It is a challenge of our time to use the power of the resources available to us for gaining knowledge and deriving from it facilitations of (human) (co)life. Possible discrimination, exclusion and exploitation as a result of technical progress at the expense of others should always be evaluated and corrected accordingly, because data analytics, whether flawed or not, have become an important tool for decision makers.

Due to the rapid change, alternative data formats are gaining in relevance over tabular data. In 1998, Merrill Lynch already said, "Unstructured data comprises the vast majority of data found in an organization, some estimates run as high as 80%." (Shilakes and Tylman, 1998, p. 15), Gandomi and Haider (2015) even speak of a share of 95%. Basically, it seems as if the share of unstructured data in the total amount of data is increasing, especially in the last years. These new forms of data require the development of new

[1] `https://www.statista.com/statistics/871513/worldwide-data-created/`, 02/25/2022

statistical and analysis methods; or rather the dependence exists the other way around. By the relatively more recently developed ability for the utilization of these data forms, unstructured data are more comprehensively collected and used for knowledge gains. Text data, which are generated in masses worldwide on a daily basis, by, e.g., newspapers, form a large proportion of this unstructured data. Digital connectivity and the resulting elimination of the need for printing have also reduced the cost of distributing text. Taking journalistic texts as an example, the increasing use of computer-generated texts, e.g., in sports coverage, shows that the supposedly largest remaining cost in text generation, the author himself, lacks existential immunity. In this loop of automated text data analysis and automated text generation, the complexity and directions of existing dependencies must be continuously examined in the long run.

1.1 Text as data

Text data are used in many domains, so there are also many research fields related to text data (Gentzkow et al., 2019). One application area that is clear to the general public is machine translation (Dabre et al., 2020). Many people are familiar with the machine translators Google Translate and, since a few years, DeepL. In addition to these (partially) commercial translators, there are a large number of open source machine translation systems developed by researchers, which are mainly evaluated using the bilingual evaluation understudy (BLEU) score (Papineni et al., 2002) because intensive human studies are too expensive. The score measures the precision of automated translation compared to human translations. This is the point at which critics have continually addressed the issue, that even for an individual person, a score of 1 would not be possible because of the inherent uncertainty in translations. Mathur et al. (2020) show that due to this uncertainty, an improvement in the BLEU score of one to two points, an order of magnitude which state of the art paper are able to achieve, comes with a real improvement in translation quality only in half of the cases. Various alternative measures have already been developed (e.g., Kocmi et al., 2021 for an extensive evaluation of automatic metrics), but these have other weaknesses, so that the BLEU score is still the established measure to be optimized in automated machine translation tasks.

The machine translation task is a special one in the field of natural language processing (NLP), because although it is a generative task, it aims at a kind of 1:1 relationship. Text summarization, in contrast, aims at condensing the information of a longer text into a shorter one. For these tailored text abstraction tasks as well as for many other NLP tasks, the corresponding leaderboards are mostly dominated by pre-trained language models based on bidirectional encoder representations from transformers (BERT, Devlin et al.,

2019) or similar transformer models. BERT is trained with the so-called masked language model. In practice, this means in the original form 15% of the words in the sentence are masked, of which again 80% are actually masked, 10% are replaced by a random word and 10% of the masked word remains original. Then, to predict the masked word, simply spoken, the model learns a kind of local sentence embedding. Typically, BERT is used in a pre-trained manner to learn basic language representations and fine-tuned to a specific wording to be used for very different NLP problems.

One of the many fundamental NLP problems on which other broader problems are built on is stance detection. In practice, this field of application is often connected to social media data (ALDayel and Magdy, 2021). In this case, for replies to tweets, the goal is to determine the writer's stance with respect to the initial tweet: Does the responder agree or disagree with the writer of the initial tweet? It turns out that there is a lot of variability in judgment for this task even among humans (Joseph et al., 2021), so evaluation is often subject to a large amount of uncertainty here as well. Reasons for the difficulty of the task, both for machines and for humans, are, e.g., irony, metaphors or fast changing word or phrase neologisms, which enjoy popularity in a strongly temporal way. An application based on stance detection is, e.g., fake news detection (Oshikawa et al., 2020). In recent years, this discipline gained attention at the latest due to rumors about intentional spreading of fake news or disinformation to manipulate elections as well as strategic information dissemination in conflicts (Jankowicz, 2020), such as happened during Russia's invasion of Ukraine in February 2022. The aim of the field of fake news detection (and closely related problems) is, on the one hand, to classify texts or statements into truth or fake, but on the other hand, it can also aim to classify seriousness, e.g., on a $[0, 1]$ scale. The combination of stance detection and rumor detection can also serve as an important desirable feature for recommender systems based on text data. As Wieland et al. (2021) show, the evaluation for news recommenders is multilayered. (Optimal) recommendations depend on the user's preferences or on the characteristics of the recommended news. Many content-based news recommenders are based on similarities of potential texts, and thus in the end aim at the creation of filter bubbles (Yao and Hauptmann, 2018). In practice, however, apart from the ethical question of whether recommender should favor filter bubbles, there are also those users who are interested in an alternative point of view on the same topic for further reading. The same applies to book recommendations, which are usually not intended to recommend exactly the same book from a different publisher. Therefore, recommender systems, just like many other NLP techniques, should not be evaluated method-based, but task-based: Most of the mentioned NLP fields have more or less established evaluation measures that need to be optimized; and mostly, the approach that the methods are evaluated task independently still dominates, although, as indicated, task specific evaluation would be more appropriate (cf. Hoyle et al., 2021).

In addition to applications primarily related to political issues, text data analysis is also useful in an economic context. Economic key indicators are published in cycles, often quarters. In the interim period between the publication of the last data and new data, there is a level of uncertainty: An uncertainty that is captured by the way of reporting in daily newspapers that can be used to predict economic indicators, or for exploration of sources of uncertainty (cf. EPU by Baker et al., 2016 and Section 3.4). A common task for the use of text data to bridging the release cycles of, in particular, survey data, is, e.g., nowcasting the gross domestic product (GDP, Garnitz et al., 2019; Thorsrud, 2020).

Topic models are used in many of the mentioned application fields. These have the advantage that the modeling idea can be explained intuitively: A set of texts is clustered into topics, whereby each text is seen as a mixture of several topics. Each topic is in turn characterized by its word distribution. This basic model idea has a natural justification without being too complex. Based on this basic idea, there are several approaches to probabilistic topic models (Blei, 2012), which in turn have been and are being adapted and optimized for many specific problems.

1.2 Latent Dirichlet allocation

The probably most well-known topic model is the classical latent Dirichlet allocation (LDA, Blei et al., 2003). In the chronology of LDA-like models, it can be understood as a further development of the probabilistic latent semantic analysis/indexing (PLSA/PLSI, Hofmann, 1999), which itself is a refinement of the latent semantic indexing (LSI, Deerwester et al., 1990). In addition, LDA can be seen as the basis for the development of the correlated topic model (CTM, Blei and Lafferty, 2007).

For the definition of the different topic models, let an observed corpus of texts be given by $\{\boldsymbol{D}^{(1)}, \dots, \boldsymbol{D}^{(M)}\}$, where a single document of length $N^{(m)}, m = 1, \dots M$ can be represented as $\boldsymbol{D}^{(m)} = \{W_1^{(m)}, \dots W_{N^{(m)}}^{(m)}\}$ and each observed word $W_n^{(m)} \in \boldsymbol{W}, n = 1, \dots N^{(m)}$ represents a realization from the set of vocabulary $\boldsymbol{W} = \{W_1, \dots, W_V\}$ of size V. In terms of topic models, for each of these observable $W_n^{(m)}$, an unobservable explanatory variable $T_n^{(m)} \in \boldsymbol{T}$ is modeled. This latent variable characterizes the topics. The user specifies a number of topics K, such that $\boldsymbol{T} = \{T_1, \dots, T_K\}$. The model then assigns a topic to each word in a document. With respect to the LDA, this procedure yields estimates for the latent word distributions $\boldsymbol{\phi}_k$ for each topic $k = 1, \dots, K$ and topic distributions $\boldsymbol{\theta}_m$ for each document $m = 1, \dots, M$. The probabilistic model of LDA

(Griffiths and Steyvers, 2004) is thus given by

$$\begin{aligned} W_n^{(m)} \mid T_n^{(m)}, \boldsymbol{\phi}_k &\sim \text{Discrete}(\boldsymbol{\phi}_k), & \boldsymbol{\phi}_k &\sim \text{Dirichlet}(\eta), \\ T_n^{(m)} \mid \boldsymbol{\theta}_m &\sim \text{Discrete}(\boldsymbol{\theta}_m), & \boldsymbol{\theta}_m &\sim \text{Dirichlet}(\alpha), \end{aligned}$$

where α and η are Dirichlet priors that control the shape of the word and topic distributions. Higher values for α and η lead to a more heterogeneous mixture whereas lower values are more likely to produce more skewed distributions. Although the LDA permits these priors to be vector valued (Blei et al., 2003), they are usually chosen symmetric because typically the user would have no a-priori information about the topic and word distributions.

In comparison to the more advanced methods, the LSI calculation only requires a frequency table of the observed words. This matrix is called document-term matrix (DTM) and indicates the M documents in the rows, the V words in the columns and the corresponding occurrence frequencies in the cells of the matrix, i.e., DTM $\in \mathbb{N}^{M \times V}$. In practice, the frequencies are adjusted by a term frequency-inverse document frequency (tf-idf) weighting before the LSI is performed, so that frequent but low-discriminant words do not get too much weight in the calculation. The tf-idf scores are calculated by multiplying the frequency values by the logarithm of the quotient of the number of documents divided by the number of documents containing this word. Based on the (mostly tf-idf weighted) DTM, the LSI is estimated by a truncated version of the singular value decomposition (SVD) by

$$\begin{aligned} &\text{DTM} = USQ^T \approx U_K S_K Q_K^T, \\ &\text{where } U \in \mathbb{R}^{M \times M}, S \in \mathbb{R}_+^{M \times V}, Q \in \mathbb{R}^{V \times V} \text{ and } U_K \in \mathbb{R}^{M \times K}, S_K \in \mathbb{R}_+^{K \times K}, Q_K \in \mathbb{R}^{V \times K}. \end{aligned}$$

Thereby, only the K largest eigenvalues and the corresponding columns of the matrices U and Q from the SVD are considered. In comparison to LSI, PLSA uses a probabilistic approach instead of SVD to tackle the problem of dimensionality reduction.

In Figure 1, five selected topic models are shown in a schematic comparison. In addition to the models PLSA, LDA and CTM, the structural topic model (STM, Roberts et al., 2013) and the author-topic model (ATM, Rosen-Zvi et al., 2004) are shown. As a refinement of the LDA, the CTM enables the modeling of correlations between topics that are simplistically treated as independent of each other in the LDA. While an LDA model predicts a word based on the latent topic that the observation suggests, CTM has the ability to predict a word associated with another topic that is correlated with the conditionally probable topic. STM can be seen as a further generalization of LDA, in the sense that it allows correlations between topics as well as the possibility to include metadata in the modeling of topic and word distributions. ATM represents the class of

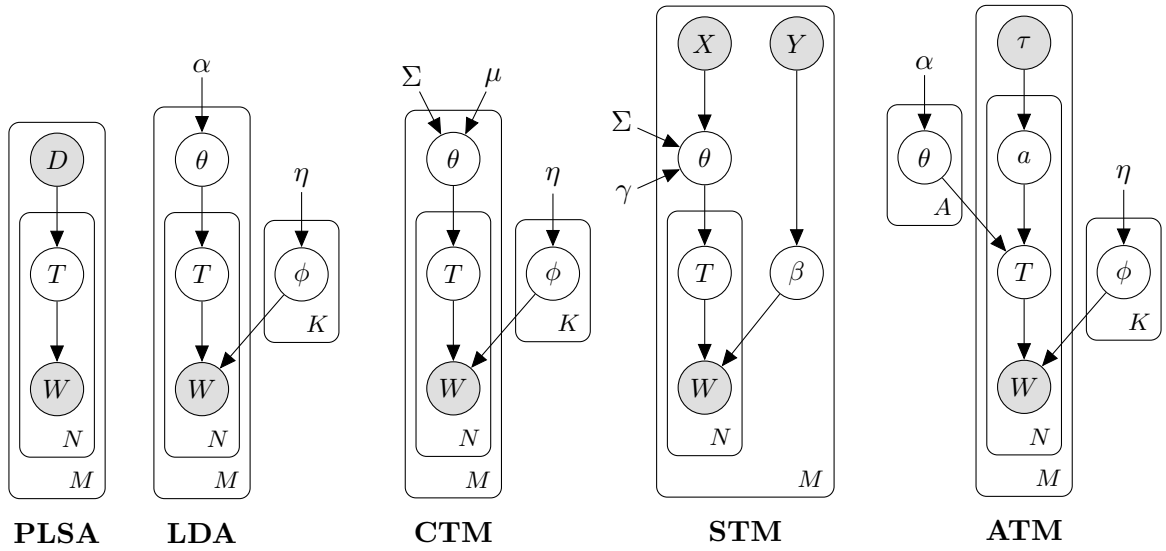


Figure 1: Schematic (plate) representation of PLSA, LDA, CTM, STM and ATM. Observable variables are marked by nodes with gray background, latent variables by plain nodes and constants are not encircled. The plates with labels M, N, K and A indicate roughly the size of the loop over which the contained variables must be iterated. In addition, parameter names are intended to indicate similar influences across models, but the relationships between the parameters are not necessarily the same as in LDA; for example, T in CTM is given by a log distribution involving the parameter θ rather than a discrete distribution as in LDA. The CTM features as an extension of the parameters of the LDA a covariance matrix Σ and a "mean" vector μ of the topics, while for the STM also the covariance matrix Σ, but here in combination with coefficients γ, as well as observed metadata X and Y affect the distributions. In the author-topic model, α denotes the prior for the topic distributions of the A authors and τ denotes the observed set of authors per document from which an author a is uniformly determined at random for each word.

task specialized topic models introduced as LDA extensions. It allows to model topic distributions per author by adding the information about the authorship of the documents and thus allows related analyses.

A major part of current research on the development of new topic models deals with neural topic models (NTM), which can be understood as a symbiosis between deep neural networks and topic models (Zhao et al., 2021). There are also approaches of combining embedding methods, such as, e.g., word2vec (Mikolov et al., 2013), with topic models, as for example the embedded topic model (ETM, Dieng et al., 2020), which aims to include word similarities in the modeling process to make the topics more coherent. Nguyen et al. (2015) show ways in which conventional topic models can be enhanced by adding word embeddings or latent feature word representations learned on external corpora, especially when modeling on smaller corpora, to improve the evidence of topics.

Each of the introduced models provides a solution in its own use cases, so that no general superiority of one of the models can be deduced. In practice, however, it can be seen that LDA, as a less complex and therefore easier to understand model, as well as flexible with

regard to the model assumptions, appears to be the most well-known and most frequently used topic model.

LDA inference is often performed using the collapsed Gibbs sampler (CGS, Griffiths and Steyvers, 2004) from the class of Markov chain Monte Carlo (MCMC) methods. Alternative estimation methods are mostly based on variational Bayes (VB), which was also proposed in the original LDA paper (Blei et al., 2003) and can be understood as an expectation-maximization (EM) algorithm. Collapsed variational Bayes inference (CVB, Teh et al., 2006), maximum likelihood estimation (ML) for PLSA (Hofmann, 2001), and maximum a posteriori estimation (MAP, Chien and Wu, 2008) are other inference methods. As a further development of the VB-based estimation methods, there is also the online variational Bayes method, which allows data not to be modeled as a whole, but to update the model for new data (Hoffman et al., 2010). In comparing the inference methods - although Blei et al. (2003) were able to show the superiority of VB over ML, as well as Teh et al. (2006) claimed the superiority of CVB over VB - Asuncion et al. (2009) discuss the different algorithms more elaborately across different parameter choices and conclude that the estimation algorithm does not have a decisive influence on the fit of the models. Rather, the authors emphasize the importance of the parameter choice and show that an appropriate choice causes the differences between the estimation algorithms in the fit to largely disappear.

In the following two sections, on the one hand an overview of different ways of evaluating topic models, especially the evaluation of LDA results (Section 2), and on the other hand the challenge of dynamic modeling of texts, mainly focusing on update algorithms for LDA (Section 3), are presented. Both sections are structured analogously: After the presentation of the current state of research, I present and explain my contributions, which I have developed together with corresponding co-authors. In this context, the corresponding implemented R package is introduced as well. After a summary of example applications in which publications I have contributed as a co-author, each section concludes with a thematically specific outlook for the corresponding discipline as a whole as well as with respect to my own research.

2 Evaluating LDA results

Evaluation of topic models is subject to intensive and continuous research. The peculiarity is that it is usually not possible to compare the models' results with "the truth". Topic models in the first place belong to the class of unsupervised methods, or might also be considered as purely descriptive methods, as they do not require a target variable for modeling. This is due to the nature of the problem that a target variable is usually not available. In the case that task-independent target variable(s) for topic models had to be defined, then there would be most likely even two, namely the true word and topic distributions of the topics and documents, respectively. Even if these were available, the definition of an evaluation measure would not be trivial due to the multi-objective optimization. In practice, however, it is still the case that not even the true distributions are available; topic models are mostly used for a way of dimension-reducing descriptive analysis of text corpora. In this case, the "target variable" to be mimicked corresponds to the human mind: A topic model should reflect as accurately as possible the human understanding of the topic structure of the corpus. This poses another problem for evaluation. The human mind is subject to variance, i.e., the same person will have different ideas of an appropriate topic structure in repeated experimental setups. In addition, there are fundamentally different opinions of an appropriate topic structure, i.e., different people can - and usually do - differ in how they perceive documents, regardless of the existing variance. For this reason, currently the most common aim in evaluating topic models is to find a measure with the highest possible correlation to the human perception of proper corpus and topic structures.

In the early years of LDA, mostly the likelihood of the models was optimized, i.e., how well the fit is with respect to the model assumptions. Perplexity as proposed in the original LDA paper (Blei et al., 2003) has been and is still widely used for this purpose. As a holdout variant, perplexity indicates how well the model can predict the words for test documents based on the modeled distributions on the training documents. Since the exact probability of held-out documents is intractable, there are other likelihood-based measures besides perplexity that can estimate this probability. Wallach et al. (2009) present two further methods, namely Chib-style estimator and left-to-right algorithm, which can predict the probability of held-out documents better and equivalently efficiently, resulting in a somewhat more accurate selection of models. However, since in the same year Chang

et al. (2009) show in a seminal work that likelihood-based measures correlate poorly, or even negatively, with human perception of "well-separated" topics, these alternatives could not gain decisive acceptance in practice. Instead, in addition to the popular perplexity, coherence-based measures, which correlate better with human judgments, have been used increasingly for the evaluation of topics, as suggested by Chang et al. (2009). For the evaluation of topic quality measures, the authors propose a human assisted procedure called "word intrusion", in which human coders are given six words, five of which have a high probability for the corresponding topic according to the topic model, and a sixth that has a low probability for the topic to be evaluated, but at the same time a high probability for one of the other topics of the model. If coders can reliably detect this "intruder word" in repeated trials, this indicates good topic quality. On this basic idea of coherent topics there are further works (e.g., Aletras and Stevenson, 2013; Mimno et al., 2011; Newman et al., 2011, 2010), in which coherence based measures are proposed, improved and evaluated, and slight adaptations of the LDA methodology are presented, which already include the new topic quality measure in the modeling. In particular, pointwise mutual information (PMI) emerges from this works as a proposal of the measure to be optimized. For two words x and y PMI is defined as

$$\mathrm{PMI}(x,y) = \log \frac{p(x,y)}{p(x)\,p(y)} = \log \frac{p(x|y)}{p(x)} = \log \frac{p(y|x)}{p(y)} \in \Big[-\infty, \min\{-\log p(x), -\log p(y)\}\Big],$$

which quantifies the difference between the probability of their joint occurrence $p(x,y)$ and the probability of their individual occurrence $p(x)$ and $p(y)$, assuming independence. In different variants the actual probability is estimated differently, e.g., via the occurrence in a sliding window, or simply via the joint occurrence in documents. Both versions have in common that they need a smoothing factor in the numerator because otherwise too many estimators would not be defined or, in practice, would produce $-\infty$ as an estimator. Stevens et al. (2012) provide a benchmark study for a number of those different coherence based topic quality measures and show that the smoothing factor has a large impact on the calculation of topic qualities. Moreover, they show that the original value, which was often selected as 1, leads to worse results and recommend much lower values instead.

As a refinement of PMI, Lau et al. (2014) propose the use of a normalized PMI (NPMI)

$$\mathrm{NPMI}(x,y) = \frac{\mathrm{PMI}(x,y)}{-\log p(x,y)} \in [-1,1],$$

for which the authors show in their comparative study that it performs best on the task of emulating human judgment on topic quality. Röder et al. (2015) address the results and show the superiority of NPMI in a comprehensive benchmark study with a number of different coherence measures. Furthermore, Xing et al. (2019) were able to confirm

that NPMI is the best performing coherence measure. In their study, they also propose a new posterior based measure named variability, which measures the variability of the estimated topic distributions across Gibbs iterations. This measure has not been used prominently in the literature, but it promises to measure another dimension of topic quality complementary to perplexity and NPMI.

Besides the quality of the resulting topics, there is (at least) one other evaluation aspect for topic models, in particular for the LDA. A weakness of the LDA is that the estimation of the model with the collapsed Gibbs sampler is stochastic and strongly depends on the initialization. Therefore, the user will get different results when running it multiple times on the same data, but different initializations. The stability of the method LDA is mostly measured by differences in the topics from different runs. These differences can be quantified with topic similarity measures. For this purpose, Aletras and Stevenson (2014) compare different topic similarity approaches, including Kullback-Leibler divergence (KLD), Jaccard coefficient (JC), cosine similarity and Jensen-Shannon divergence (JSD). In addition, the authors also consider a PMI based approach in which topic similarity is determined by computing the average pairwise PMI between the topic words in two topics. Accordingly, the authors show that the aforementioned classical word probability measures perform weaker in terms of correlation to the human perception of topic similarity (measured on a scale from 0 to 5) than, in particular, the PMI based approach. However, the literature still mainly uses probability based measures. Greene et al. (2014) use an average Jaccard coefficient (AJC) that considers top word sets up to different depths and averages the different resulting values. The authors use this measure for tuning the number of topics based on the stability of the models over multiple runs. Similarly, Su et al. (2016) use the same definition of AJC to quantify the stability of LDA models on noisy texts. While Agrawal et al. (2018) compute stability as the median of JCs of the top 9 words, Mantyla et al. (2018) use the so-called rank-biased overlap (RBO) to compute the stability of LDA models over replicated runs. Thereby, similar to the AJC, it compares top word lists of different depths and additionally weights deeper lists less strongly using a hyperparameter. In addition, Maier et al. (2018) use cosine similarity in their best practice guide and match topics that exceed a threshold, and Morstatter and Liu (2018) propose a measure that is supposed to optimize interpretability complementary to coherence and refer to it as consensus. Each of these measures can be used in different variations to measure LDA stability.

In order to not only measure the weakness of the LDA, the instability, but to improve it, Nguyen et al. (2014) propose to average Gibbs iterations. For this purpose, after a burn-in period, for example, the Markov chain is stored every 20 iterations; let us assume for a total of 200 iterations, which results in 10 states. Starting from these 10 chains, states are stored every 10 iterations and these Markov chains run for a total of 50 iterations. In total,

using these hyperparameters, this results in $10 \times 5 = 50$ estimated probability matrices $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$, which can be averaged. This reduces the variance of the result because not only a single state (usually the last one after a fixed number of iterations) is used, but movements between topics are captured by the means. One disadvantage of this method is that the - to be honest highly uncertain - assignments of each token to topics are lost. In addition, this method is also based on only a single initialization. This problem could be solved if models of different initializations could be averaged. However, this is not trivial, since it is not clear how topics of models of different initializations can be - and to what extent can be - matched. Among others, this task has been addressed by Maier et al. (2018). The authors propose to perform the initialization reasonably using the co-occurrences and to make the results of the LDA more reliable by using cosine similarity and topic matching of topics that realize a pairwise similarity above a threshold. Topics that do not achieve similarity above this threshold are classified as unstable and are not considered further. This method is again based on a single initialization and manipulates the results of the LDAs to a degree that it is unclear how this restriction to stable topics affects conclusions drawn from these results. An attempt to stabilize the model of the LDA by adapting the modeling was made by Koltcov et al. (2016) with the granulated LDA (GLDA). Here, the authors consider sliding windows with typical kernel functions (step, triangular, Epanechnikov) within which the assignments to topics are set identically. This adaptation of the Gibbs sampler makes the results more stable, but was only presented by the authors on one (small) dataset and a publication of the implementation is pending.

One procedure that, to the best of my knowledge, has not yet been used to overcome the instability of LDA is to select a model from a set of repeated runs that differ only in their initialization, using a criterion that measures the extent to which the different outcomes are similar to a potential median outcome. The selection algorithm LDAPrototype, which I present in the following, builds on this approach.

2.1 Contributed publications

Methods

Rieger, Jonas (2020). “ldaPrototype: A method in R to get a Prototype of multiple Latent Dirichlet Allocations”. In: *Journal of Open Source Software* 5.51, p. 2181. DOI: 10.21105/joss.02181.

Rieger, Jonas, Carsten Jentsch, and Jörg Rahnenführer (2020a). “Assessing the Uncertainty of the Text Generating Process Using Topic Models”. In: *ECML PKDD 2020 Workshops*. Vol. 1323. CCIS. Springer, pp. 385–396. DOI: 10.1007/978-3-030-65965-3_26.

Rieger, Jonas, Carsten Jentsch, and Jörg Rahnenführer (2022a). "LDAPrototype: A Model Selection Algorithm to Improve Reliability of Latent Dirichlet Allocation". In: *Preprint available at Research Square*. DOI: 10.21203/rs.3.rs-1486359/v1.

Rieger, Jonas, Jörg Rahnenführer, and Carsten Jentsch (2020b). "Improving Latent Dirichlet Allocation: On Reliability of the Novel Method LDAPrototype". In: *Natural Language Processing and Information Systems, NLDB 2020*. Vol. 12089. LNCS. Springer, pp. 118–125. DOI: 10.1007/978-3-030-51310-8_11.

Applications

Jentsch, Carsten, Enno Mammen, Henrik Müller, Jonas Rieger, and Christof Schötz (2021). "Text mining methods for measuring the coherence of party manifestos for the German federal elections from 1990 to 2021". In: *DoCMA Working Paper* #8. DOI: 10.17877/de290r-22363.

von Nordheim, Gerret and Jonas Rieger (2020). "Im Zerrspiegel des Populismus – Eine computergestützte Analyse der Verlinkungspraxis von Bundestagsabgeordneten auf Twitter". In: *Publizistik* 65. German. [Distorted by Populism – A computational analysis of German parliamentarians' linking practices on Twitter], pp. 403–424. DOI: 10.1007/s11616-020-00591-7.

von Nordheim, Gerret, Jonas Rieger, and Katharina Kleinen-von Königslöw (2021). "From the Fringes to the Core – An Analysis of Right-Wing Populists' Linking Practices in Seven EU Parliaments and Switzerland". In: *Digital Journalism*, pp. 1–19. DOI: 10.1080/21670811.2021.1970602.

2.2 Model selection via LDAPrototype

As described in Section 2, there are different criteria for model evaluation for a set of LDA models. Stability describes the similarity of R repeated runs of the model, while reliability does not measure the stability of the model, but the similarity of the results of H repeated executions of a procedure on different sets of LDA models of size R. The quality of an LDA result, in contrast, can usually only be assessed with external knowledge combined with a choice of an optimization criterion. In Figure 2, the different terms, and their determination, are shown schematically.

Instead of avoiding the instability of LDA by adapting the method or via optimization, the instability can also be understood as an algorithmic variable that can be averaged out in a statistical sense. The LDAPrototype method follows the classical statistical approach of running the procedure repeatedly and, in a sense, using the mean run. In Rieger et al. (2022a, cf. page 71) we explain step by step the motivation and definition of the similarity measure similarity of multiple sets by clustering with local pruning (S-CLOP) with which we measure similarities between LDA models. In this case, two LDA models are completely

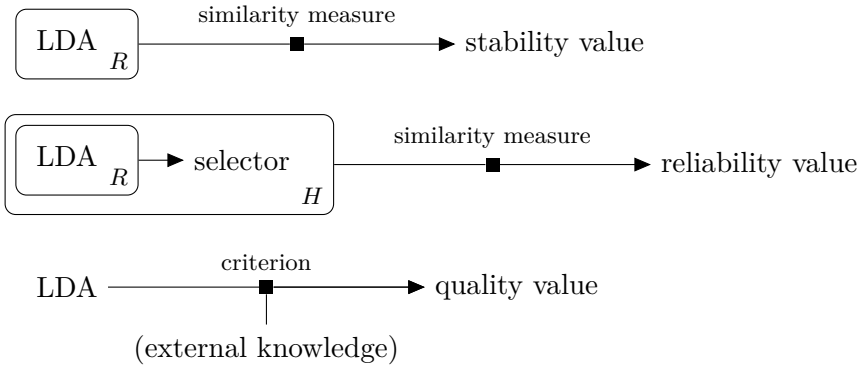


Figure 2: Schematic explanation of terms stability, reliability and quality.

$$
\begin{array}{c|cccc|ll}
 & \mathrm{LDA}_2 & \mathrm{LDA}_3 & \dots & \mathrm{LDA}_R & & \\
\hline
\mathrm{LDA}_1 & s_{12} & s_{13} & \dots & s_{1R} & \bar{s}_1 & = \frac{1}{R-1}\sum_{j=2}^{R} s_{1j} \\
\mathrm{LDA}_2 & & s_{23} & \dots & s_{2R} & \bar{s}_2 & \\
\mathrm{LDA}_3 & & & & s_{3R} & \bar{s}_3 & \\
\vdots & & & \ddots & \vdots & \vdots & \vdots \\
\mathrm{LDA}_{R-1} & & & & s_{R-1;R} & \bar{s}_R & = \frac{1}{R-1}\sum_{j=1}^{R-1} s_{Rj}
\end{array}
\quad
\begin{array}{l}
\mathrm{opt} := \underset{i\in\{1,\dots,R\}}{\arg\max}\ \bar{s}_i \\
\Rightarrow \mathrm{proto} = \mathrm{LDA}_{\mathrm{opt}}
\end{array}
$$

Figure 3: Schematic representation of the determination of a prototype based on a set of LDA models.

similar if, for each topic from one LDA, the clustering procedure finds a matching topic from the other LDA that is not more similar to any topic from its own LDA than to the potential partner topic from the other LDA. Thus, a similarity of 1 says nothing more about the matched topics than that each of the topics from the one LDA run finds a reliable counterpart in the other run, but in particular not how similar the matched topics actually are. The similarity measure captures more the similarity of the models' topic structures than the similarity of the actual topics.

For this reason, the measure is well suited for selecting from a set of potential candidates the LDA that is on average most similar to all others in terms of structure. The LDAPrototype selection algorithm determines a kind of medoid of the models, which is shown schematically Figure 3. Thus, from the pairwise similarities $s_{ij}, i \neq j$, the mean similarities $\bar{s}_i$ are calculated for all runs $i = 1, \dots, R$ and the maximum is determined. The LDA that realizes the largest similarity then becomes the PrototoypeLDA. In Rieger et al. (2020b, cf. page 123) we show that this procedure strongly improves the reliability of the selected LDAs compared to random selection - which is still commonly performed in the literature. In the study, we use the almost 3500 non-empty texts from April 2019 in the Süddeutsche Zeitung with three different numbers of topics $K = 20, 35, 50$ and 25 000 LDAs for each K, i.e., 75 000 LDAs in total. We show that the similarity of LDAs selected using the LDAPrototype algorithm are considerably higher than those of random selected LDAs.

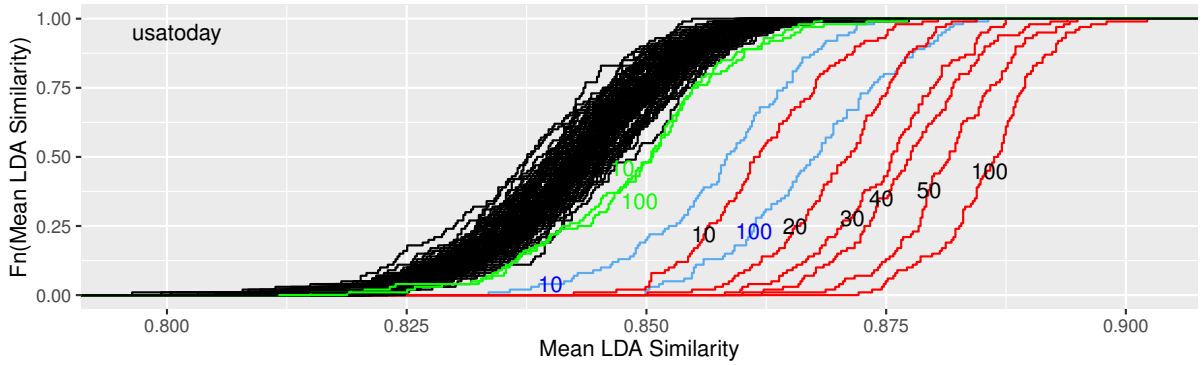


Figure 4: Increase of reliability in dependence of the number of replications R on a dataset of all articles in the newspaper USA Today from June to November 2016: ecdfs of the mean similarities calculated on $R = 100$ replications of randomly selected LDA runs (black) and on the $H = 100$ most representative prototype LDA runs (red) based on subsamples of $R = 10, 20, 30, 40, 50$ or all 100 LDA runs. For comparison, the ecdfs of the 100 selected LDA runs using perplexity (blue) and NPMI (green) based on the subsamples of 10 runs and all 100 runs are given.

In a broad study, we demonstrate in Rieger et al. (2022a) that the use of LDAPrototype increases reliability - measured as mean LDA similarity - more than selection by perplexity or NPMI. This is shown in Figure 4 visualizing a comparison of the empirical cumulative distribution functions (ecdfs) of the calculated mean LDA similarities. The selection via LDAPrototype highers the basis reliability score of 0.84 to a reliability score of 0.89, while perplexity and NPMI realizes scores of 0.87 and 0.85, respectively. It is noticeable that the LDAPrototype method already provides higher reliability for $K = 20$ candidate models than selection via perplexity from $R = 100$ models. In the study, we intensively compare different choices of $R = 10, 20, 30, 40, 50, 100$, and different similarity measures for determining topic similarities (cosine, thresholded JC, JSD, RBO); and we also evaluate the increase in reliability for five different datasets of different origins (e.g., Twitter and newspapers). We were able to show that the method highly increases the reliability for all datasets considered. Using a dataset of nearly two million articles from the New York Times and $K = 100$, we demonstrate computability and usability even for large corpora. Nevertheless, we see the greatest practical relevance in applying the method in the case of smaller datasets. We were able to show that for large datasets the variability of the models is basically not that great, so that such a high increase in reliability as for smaller datasets is not possible for larger ones.

The application of the LDAPrototype method leads in practice to the fact that users do not have to worry about the suitable choice of an LDA model for a fixed parameter combination $\{K, \alpha, \eta\}$, as, e.g., by manual coding or eyeballing. Instead, the selection algorithm automatically selects a reliable representative LDA, the medoid from a set of R LDAs. Usually, the reliability increases strongly already for $R = 50$. In general, we recommend the use of 100 candidate models, as far as computability is given. For

particularly small corpora, it may be reasonable to choose a significantly larger number, e.g., $R = 500$.

2.3 R package ldaPrototype

The methodology of determining a PrototypeLDA is implemented in the R package ldaPrototype. In Rieger (2020, cf. page 43), the package is briefly described and set in context to other common text processing tools in R. The package has been tested for desired functionality and peer-reviewed, as well as published via CRAN, and provides extensive documentation on GitHub and via its help pages. The computation of the LDA models is based on the implementation of the R package lda (Chang, 2015). Parallel calculation is achieved by linking to parallelMap (Bischl et al., 2020) and for cluster calculations by linking to batchtools (Lang et al., 2017).

Usability and robustness are enhanced by object-oriented programming. For example, the package provides classes - and, for instance, corresponding `print` functions - for `LDA`, `LDARep`, `LDABatch`, and `PrototypeLDA` objects. By means of suitably implemented `getter` functions, the results of the calculations can also be used for further processing in the package tosca (Koppers et al., 2020).

2.4 Example: Political parties' linking practice

The LDAPrototype selection algorithm has already been applied in several publications. In von Nordheim and Rieger (2020), we analyze over half a million tweets, namely the tweets of all parliamentarians of the corresponding legislative period. We visited all of the more than 132 thousand shared URLs and used a tool to extract the texts of links to journalistic content. The results show that individual AfD politicians apparently share thematically more broadly than the average parliamentarian, but at the same time the overall agenda of all AfD parliamentarians is less diverse compared to the other parties.

In a further study (von Nordheim et al., 2021), we extended the analysis to eight European parliaments. In Figure 5, parties are indicated by their party abbreviations and color-coded by country. In addition, the far-right parties are marked according to their integrity in the respective parliament. It can be seen that source and topic insularity do not measure the same thing. In many studies today, however, topic diversity is still quantified by the number of different sources. Our analysis shows that this approach is not appropriate. The different dimensions measure different patterns and can be interpreted appropriately. A positive deviation from the mean topic insularity, i.e., an increased topic insularity, occurs

Figure 5: Scatter plot of deviations from topic and source insularity from their countries' mean value for all parties. Negative integrated right-wing populist parties (Neg.-integr. RWP) are shown as a triangle, non-integrated right-wing parties (non-integr. RWP) as a rectangle; all other parties as a circle. Only those parties are shown that received more than 9.5% of the LDA assignments in the respective country. The parties are colored according to their country and sized according to their share of assignments in that country. For a list of all party abbreviations, see the appendix of von Nordheim et al. (2021).

mainly for right-wing parties. Six out of eight observed parties with a deviation greater than zero are right-wing parties and the threshold only misses VOX as Spanish right-wing party. In addition, the source insularity can then be used as an indicator of how well such a right-wing party is integrated into parliament. The non-integrated parties PVV and AfD have a positive source insularity, i.e., they refer to more content from kind of partizan media than other parties. In the study, we were able to show that even right-wing parties that are solidly integrated in their countries, such as the FPÖ in Austria, tend to share content in a more insular fashion on Twitter compared to other parties in the same country.

In addition, in the working paper by Jentsch et al. (2021), we use the LDAPrototype to examine which topics are addressed in the party programs of the German Bundestag parties for the 2021 federal elections. More specifically, we investigate the intensity with which the parties discuss issues and which vocabulary they use for it. It turns out that, measured by the written documents in the run-up to the election, the Kenya and Traffic Lights coalitions are the ones with the highest matching agendas.

In the three example applications described, an analysis without the LDAPrototype method would have involved intensive extra work in selecting a suitable LDA. Even if the

presented applications of the method are mainly in the field of political sciences so far, an application in other areas is equally conceivable.

2.5 Outlook: Task based topic model evaluation

As shown in Figure 2, there are different ways of topic model evaluation. Besides the quality of the results, reliability and stability are of interest. We distinguish between reliability and stability to make clear that an imposed procedure for similarizing the results of a workflow that does not manipulate the methodology itself does not affect the stability of the model, but in this case improves the reliability of the results. Just looking at the stability of the models, there are already differences which uncertainty to measure. Besides the algorithmic uncertainty, which is often of interest, there is also the aleatoric uncertainty, which results from the text generating process of the model (Benoit et al., 2009). In Rieger et al. (2020a), we investigate the influence of the latter uncertainty, whereas Section 2 offers an outline of methods that address the instability of the LDA.

It has been shown that the selection algorithm via LDAPrototype improves the reliability of the results. To what extent this improvement represents a qualitative improvement of the results compared to random selection, remains unanswered so far. In principle, it can be assumed that a medoid model selected by LDAPrototype tends to be less prone to artifacts and mismodeling than a single randomly selected model. Thus, selection by LDAPrototype is likely to improve the quality in terms of robustness of the results.

Further, it is not entirely clear how quality can be interpreted in the context of topic models. On the one hand, it is certainly true that there are application areas in which a particularly high correlation of the model with the human perception of well-divided topics is the criterion of interest to be optimized. This could be of particular relevance in the context of recommender systems. Human coder experiments are necessary for this type of evaluation. Proxy methods that correlate particularly well with human sentiment can be applied, but require ongoing monitoring that they continue to provide the desired proxy properties (cf. Hoyle et al., 2021). On the other hand, there are many application areas in which evaluation based on the mentioned approach is certainly not the criterion to be optimized. An example is the use of topic models in the workflow of a classification problem. In this case, the variable to be optimized is the evaluation measure of the classification problem, often the misclassification rate. Already in 2013, Grimmer and Stewart (2013) suggest in their guideline that a task-specific evaluation might be useful. In their "four principles of quantitative text analysis", they point out that first, "all quantitative models of language are wrong - but some are useful", second, "quantitative methods augment humans, not replace them", third, "there is no globally best method for

automated text analysis", and fourth, they encourage "validate, validate, validate". These principles suggest that there is no generic way of evaluating topic models. Rather, the evaluation needs to be appropriately adapted to the problem and the resulting optimality criterion for each individual task.

Ding et al. (2018) show, for example, that while many topic models optimize perplexity, topic coherence measured by NPMI does not correlate strongly with perplexity, but rather the optima are already achieved in earlier iterations of the modeling. For this reason, the authors give an example of the potential of integrating coherence awarenes into NTMs so that during modeling itself, the corresponding optimality criterion - in this case NPMI - is optimized directly. Nguyen et al. (2015) also show that the addition of latent feature representations, which can be learned on external corpora, can improve the coherence of the topics of conventional topic models.

Regarding multi-objective optimization of topic models, Ethayarajh and Jurafsky (2020) note that the typical NLP leaderboards generate optimal results to some extent, but that practical utility is more important than an artificially generalized optimality criterion. In this context, there is often a discrepancy with the leaderboards and, in particular, there is no one-fits-it-all solution, but the authors also favor task-specific optimization. Doogan and Buntine (2021) show in this context that coherence measures can be unreliable for specific tasks. The authors show this with the example of Twitter data. Hoyle et al. (2021) confirm the results and note that new models like NTMs need new coherence measures. The existing measures are aligned with the architecture of conventional topic models and therefore do not provide good proxy properties for substituting human coder experiments. Hence, the authors emphasize that the evaluation of topic models understanding them as proxies of human judgments as well as the evaluation of human judgments themselves need to be reassessed and they suggest a task-based evaluation instead of a model-based evaluation.

Furthermore, model parameter tuning is a field of research that can be explored more intensively and reliably based on mature evaluation methods. Thus, if the criterion/criteria for measuring model goodness is/are clearly defined, parameter tuning of the set $\{K, \alpha, \eta\}$ can be performed for LDA, for example. Accordingly, criteria for assessing model goodness should not be systematically biased in the choice of "best" parameters.

Based on this, my goal for further research is to create a set of criteria for automated and human assisted topic model evaluation, which are to be optimized at the same time and to normalize and weight them appropriately, so that for each type of task a linear combination of the criteria becomes the evaluation measure to be optimized in the sense of task-based leaderboards. For specific applications in practice, the weighting of the criteria should, of course, be adapted individually. However, a comparison on leaderboards

requires a clear ranking, so that an unweighted side-by-side comparison of the different criteria does not seem satisfactory and viable in practice.

3 Updating and monitoring LDA results

Information from text is often used, among other scenarios, when other data sources are not available until a later point in time. In these cases, the models must be able to be calculated fast. It is helpful if the model has the possibility to update previous statuses with new data in such a way that the modeling of the earlier documents is not influenced and thus a temporal consistency of the results is ensured. In addition, there are many applications for monitoring text corpora, for which update algorithms that allow temporal changes of modeled topics are also essential. There are some methods that already aim to model temporal topics or allow updates of the model.

With topics over time (TOT), Wang and McCallum (2006) propose a continuous generative model that assigns timestamps to every single word in all documents. Thus, a prediction of the publication time of each document is possible. In particular, the model is capable of creating temporally narrow topics based on the knowledge that specific word co-occurrences appear very frequently in a short period of time. For modeling the past, this model also uses the future. Furthermore, updating the model with new data is not possible or reasonable without adapting the existing model. Therefore, TOT must be understood as a snapshot model that can be updated in a meaningful way only by recalculation. The continuous time dynamic topic model (cDTM) by Wang et al. (2008) implements a very similar modeling idea. It is a continuous version of the discrete time dynamic topic model (dDTM) by Blei and Lafferty (2006), which also does not implement a classical update algorithm. Instead, the topic evolution is integrated into the modeling with the help of parameters, so that future texts have an influence on the modeling of previous texts in dDTM as well. Using the outbreak of Covid-19 in early 2020 as an example, it can be shown that even a few months before the first occurrence of the word *covid* in the dataset, the estimated probability of its occurrence increases sharply. In particular, this means that when new data are added, this model also changes for past time points, violating the temporal consistency of the result. Rather, dDTM and cDTM are appropriate for a setting in which smoothing the temporal topic distributions as part of a single snapshot model is of interest.

Hong et al. (2011) propose a model with local and shared topics in text streams. The authors include the prevalence of the different topics in the modeling and focus on

simultaneous modeling of tweets and news texts to identify local topics. This model is also not explicitly built as an updating algorithm, but as a snapshot model that incorporates temporal structures. With temporal LDA (TM-LDA), Wang et al. (2012) present an approach to model text streams - especially many short texts by the same author - using transition matrices. The authors themselves consider the main application of the model to be in the area of social media posts because a large number of texts per user over the entire time period should be available for (meaningful) modeling. The focus is on temporal modeling of topics per author instead of global changes of topics. In contrast, Streaming-LDA (Amoualian et al., 2016) models dependencies between consecutive documents based on Dirichlet distributions or copula based. Consecutive modeling means that the order in which documents are considered has a large impact. Dependencies between documents that do not occur consecutively can thus only be modeled implicitly via the intervening documents. Due to the repetition of Gibbs iterations, information of subsequent documents is used to model previous documents in this model as well. As mentioned, this is an unrealistic situation in many practical application fields that are based on monitoring.

All described models can only deal with fixed vocabularies or do not address how they could deal with new vocabularies. This is a disadvantage for real monitoring scenarios in which the existing model is updated over a longer period of time repeatedly, but old states of the model - as well as, e.g., time series based on the results of the topic model - are to be kept consistent. Thus, a recalculation of the model is neither intended nor practicable in this application scenario. Zhai and Boyd-Graber (2013), in contrast, propose a variant of an online LDA that considers an infinite vocabulary, i.e., allows adding new as well as deleting no longer used vocabularies. The model considers documents in minibatches, so it models new documents with the knowledge of the entire past. Due to this design, the model is suitable for use in a monitoring context, but it is likely that abrupt changes in topics cannot be modeled, and even new topics cannot be created, since online LDA, as a relatively static model, takes the entire past into account.

For this reason, there is an ongoing need for research in the area of update algorithms for LDA. In particular, flexible and at the same time simple models that can be used in the monitoring domain are of special interest to users. In Section 3.2, I present our RollingLDA method, which uses minibatches to ensure that adding new data do not change the assignments of old documents. In addition, the method provides possibilities for intuitive parameter customization, allows the addition of new vocabularies and the evolution of existing topics. Thus, the method is well applicable in the monitoring context.

In the field of monitoring, the goal is often to identify events, structural changes, narrative shifts, or changes in topic or word distributions. For this purpose, topic similarity measures

are mostly used. In contrast to the stability measurement of LDA models (cf. Section 2), where mostly similarities between different topics from different models are calculated, in this scenario the similarity of the same topic at different points in time is determined. For this purpose, often other similarity measures than those mentioned in Section 2 are appropriate.

Keane et al. (2015) use cosine similarity to determine so-called eventy topics. The authors model a single LDA model for each day of interest, which also contains the data of the last nine days. Then, topics at different points in time are matched using the maximum cosine similarity. The authors use the time series of similarities of a single topic to infer whether it is an eventy topic based on noticeably low cosine similarities or a non-eventy topic. They also note that identification work similarly well with JSD in their scenario. The assumption that for each topic there is a matching counterpart in the next day's LDA model is already bold, but could still be justified by the rolling data processing. However, it seems unlikely that the same topic can be reliably found in every daily LDA model over a longer period of time. Thus, the impact on the interpretation of the results of the identified eventy topics is difficult to assess. Similarity calculation via JSD is also used by Xu et al. (2019) in their study on the evolution of topics in news data. They show that LDA is well suited as a method to detect structural changes in topics of news data. Wang and Goutte (2018) also use LDA models and compare cosine similarity and JSD in combination with different change point algorithms as well as standard LDA and online LDA in their study. They found out that online LDA (cf. Zhai and Boyd-Graber, 2013) performs on par with standard LDA for the task of detecting change points.

A lot of methods are limited to identifying often only one single change point in the offline scenario. For example, Bose and Mukherjee (2021) propose the use of LDA to determine offline changes in the topic distribution of texts. At the same time, there are works based on Bayesian online monitoring. Kim and Choi (2015) present a method they name document-based Bayesian online change point detection, in which they present a generative model for word frequencies and word impacts. They improve the detection algorithm using a regression approach, so that the model learns which words frequently show conspicuous behavior close to identified changes, i.e., which words seem to be responsible for changes. While in the mentioned methods changes in the global topic distributions of documents are considered as changes, Frermann and Lapata (2016) consider changes in topic distributions of words. For this, they use a dynamic Bayesian model to localize diachronic meaning change. Liang and Wang (2019), however, investigate changes of sentiment in topics.

In comparison to the previously mentioned related methods, some of which consider global changes in topic distributions of texts (cf. Bose and Mukherjee, 2021; Keane et al., 2015; Kim and Choi, 2015), sentiments in topics (cf. Liang and Wang, 2019) or changes in

topic distributions of words (cf. Frermann and Lapata, 2016), I will briefly explain in Section 3.5 how we use RollingLDA to detect changes in word distributions of topics over time. Thus, we show that RollingLDA can also be applied well in the context of monitoring. In addition, in Section 3.4 I will explain how we use the method to create time consistent time series of indicators based on newspaper articles.

3.1 Contributed publications

Methods

Rieger, Jonas (2021). *rollinglda: Construct Consistent Time Series from Textual Data*. R package version 0.1.0. DOI: 10.5281/zenodo.5266717. URL: https://github.com/JonasRieger/rollinglda.

Rieger, Jonas, Carsten Jentsch, and Jörg Rahnenführer (2021). "RollingLDA: An Update Algorithm of Latent Dirichlet Allocation to Construct Consistent Time Series from Textual Data". In: *Findings Proceedings of the 2021 EMNLP-Conference*. ACL, pp. 2337–2347. DOI: 10.18653/v1/2021.findings-emnlp.201.

Rieger, Jonas, Kai-Robin Lange, Jonathan Flossdorf, and Carsten Jentsch (2022b). "Dynamic change detection in topics based on rolling LDAs". In: *Proceedings of the Text2Story'22 Workshop*. Vol. 3117. CEUR-WS, pp. 5–13. URL: http://ceur-ws.org/Vol-3117/.

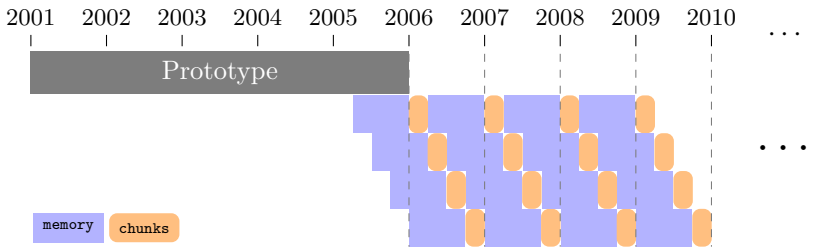
Applications

Müller, Henrik, Jonas Rieger, and Nico Hornig (2022a). "Vladimir vs. the Virus – a Tale of two Shocks. An Update on our Uncertainty Perception Indicator (UPI) to April 2022 – a Research Note". In: *DoCMA Working Paper* #11. *Previous versions:* "Riders on the Storm" (Q1 2021), "We're rolling" (Q4 2020), "For the times they are a-changin'" (Q3 2020). DOI: 10.17877/DE290R-22780.

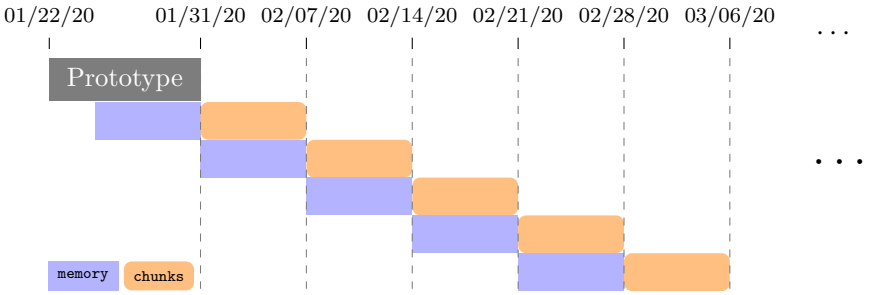
Müller, Henrik, Jonas Rieger, Tobias Schmidt, and Nico Hornig (2022b). "Pressure is high – and rising: The Inflation Perception Indicator (IPI) to 30 April 2022 – a Research Note Analysis". In: *DoCMA Working Paper* #10. *Previous versions:* "A German Inflation Narrative" (02/28/2022). DOI: 10.17877/DE290R-22769.

3.2 Rolling approach via RollingLDA

The demand for update algorithms from the area of application, especially in the field of monitoring, is high (cf. Section 3). At the same time, models in this field should satisfy some properties to be practically applicable. First, the model should be updatable without



(a) Rolling strategy for modeling newspaper articles to extract the UPI (Müller et al., 2022a); chosen parameters: `chunks = "quarter"`, `memory = "3 quarter"`, `init = "2005-12-31"`, `type = "ldaprototype"`. Note: Since the beginning of 2022, the UPI is updated monthly, i.e., `chunks = "month"`.



(b) Rolling strategy for modeling online articles to monitor changes in Covid-19 related news (Rieger et al., 2022b); chosen parameters: `chunks = "week"`, `memory = "week"`, `init = "2020-01-31"`, `type = "ldaprototype"`.

Figure 6: Schematic representations for different rolling strategies.

recalculation. On the one hand, this is because updating needs to be computed quickly in many cases. On the other hand, previous results should remain valid and thus a temporal consistency of the results derived from the LDA models should be ensured. Moreover, an update algorithm should be easy to use and allow flexibility of the model by intuitive parameters while preserving a high reliability of the model.

RollingLDA (Rieger et al., 2021, cf. page 59) implements an update algorithm of classical LDA that allows the construction of temporally consistent time series. The method considers the texts to be modeled in chunks or minibatches and models them in a sequential manner. First, a comparatively large starting batch is used as the initial chunk. All texts within this initialization period are modeled by default via LDAPrototype (cf. Section 2.2) to create the starting topics as reliably as possible. Initialization via classical LDA is also possible. After initialization, the texts are modeled sequentially in chunks and a proportion of the modeled previous texts is used as initialization for the topics in the corresponding chunk. Pseudocode of the implementation can be seen in Algorithm 1 on page 62.

The method offers the user the possibility to flexibly adapt the modeling via parameter settings. The choice of the parameters can be made intuitively by a-priori knowledge or practical experience. The most important parameters are `chunks`, `memory` and `init`. In Figure 6 the parameter choices of two example applications (cf. Section 3.4 and 3.5) are shown schematically. By specifying `chunks`, the frequency of modeling can be set,

i.e., how often new texts are integrated. In Figure 6 these texts are displayed in orange. If this parameter is chosen as `"week"`, for example, seven consecutive days of texts are always collected and then modeled together as a minibatch. In doing so, not the complete knowledge of the model is used for initialization of the topics of the minibatch, but only texts that are within a time period specified by `memory` (highlighted in blue in Figure 6). Specifically, a choice of `"3 week"` means that the last three weeks before the start of the current minibatch will be used to initialize the topics. The `memory` parameter should be chosen with respect to prior knowledge of seasonally existing topics, as the model will completely lose awareness of a topic if it does not appear in the entire memory. In addition, `init` determines the date up to which the first initial chunk lasts. Up to this date all texts are modeled together. In Figure 6 this complete period is colored gray. For this, a (usually much) larger time period than `chunk` is chosen to ensure reliable starting topics for further modeling. The method offers the possibility of integrating further parameters, but these are not specific for RollingLDA, but usually have to be chosen in some way in LDA oriented modeling (cf. Section 3.3).

RollingLDA allows a simple use without tuning due to the possibility of intuitive parameter choices and is also well suited for the monitoring context due to the non-necessary recalculation of the model. By including the LDAPrototype method, a reliable modeling of the topics can be ensured, while the rolling approach makes the modeling flexible enough to allow for an evolution of the topics. Thus, the model allows new topics to evolve and is able to discard non-prevalent topics. At the same time, a time consistent mapping of the same topic is guaranteed, so that based on this, monitoring of changes within the topics becomes possible - without the need to perform a matching of topics from different LDA runs beforehand.

3.3 R package rollingIda

The R package rollingIda (Rieger, 2021) implements the RollingLDA methodology. The code for modeling is based on the C code of the R package lda. The code has been cleaned up and modified to enable updates by appropriate parameter setting according to the systematic explained in Section 3.2. Furthermore, the object-oriented programming builds on the objects from ldaPrototype (cf. Section 2.3) and adds `getter` and `print` functions for `RollingLDA` objects. Within the object, the preprocessed documents are automatically stored in a way that is safe for the user, so that in each case a direct mapping to the corresponding publication date, chunk association and topic assignments is easily possible. Using `getChunks` the user obtains information about the modeled chunks in a tabular form. In addition to the start date, end date and the start date of the memory period,

the number of modeled documents in the corresponding chunk, the number of discarded documents within the preprocessing (due to different thresholds) as well as the number of documents in the memory and the size of the vocabulary in this chunk are given. Here, the size of the vocabulary is monotonously increasing, since words once included are always taken into account in later chunks.

The package provides plausible default parameters, such as for the thresholds for the size of documents and frequencies of vocabularies to be included, to allow for fast and easy usage. Moreover, the two main method parameters `chunks` and `memory`, which do not have a default, can be passed in a user-friendly and flexible way, that is as a vector of `dates`, a single `integer` or a single `character` (e.g., `"week"` or `"2 month"`). Furthermore, by linking to the object-oriented style of ldaPrototype, the objects are also directly usable for the functions of tosca by means of respective `getter` functions.

3.4 Example: Text based indicators

The rolling approach of the method offers a broad range of possible usage in true application areas, especially in the field of monitoring. An example application is the uncertainty perception indicator (UPI, Müller et al., 2022a), which is used to measure the proportion of coverage of various uncertainties in the three German newspapers *Die Welt*, *Handelsblatt* and *Süddeutsche Zeitung*. For this purpose, we build a subcorpus of the complete set of all 2.9 million published articles from 01/01/2001 to 04/30/2022. This is created by a proper search term combination to find those articles dealing with economic uncertainty which results in 39 058 texts. The resulting subcorpus is modeled using RollingLDA. The initialization period is chosen to end on 12/31/2005; all articles published up to that date are modeled using the LDAPrototype method. From then on, the model is updated monthly with the newly published articles using the articles from the three quarters before as memory. In Figure 6a, a modeling scheme of an earlier version of the UPI is shown as an example. It differs only in the parameter `chunks` as until 2022 the UPI was updated quarterly instead of monthly.

Uncertainty and its economic impacts have gained plenty of attention in recent years, especially after the financial crisis of 2008 and again since the rise of populist politics. The best-known indicator for measuring uncertainty and making its economic impact more predictable is the economic policy index (EPU, Baker et al., 2016). An advantage of the UPI compared to the EPU is the possibility to also specify the topical sources of uncertainty and to observe their trend over time. We model a total of 14 topics for the UPI using RollingLDA. One of the topics is composed mainly of the false positives of the search term, i.e., the articles that satisfy the search term but are not about economic

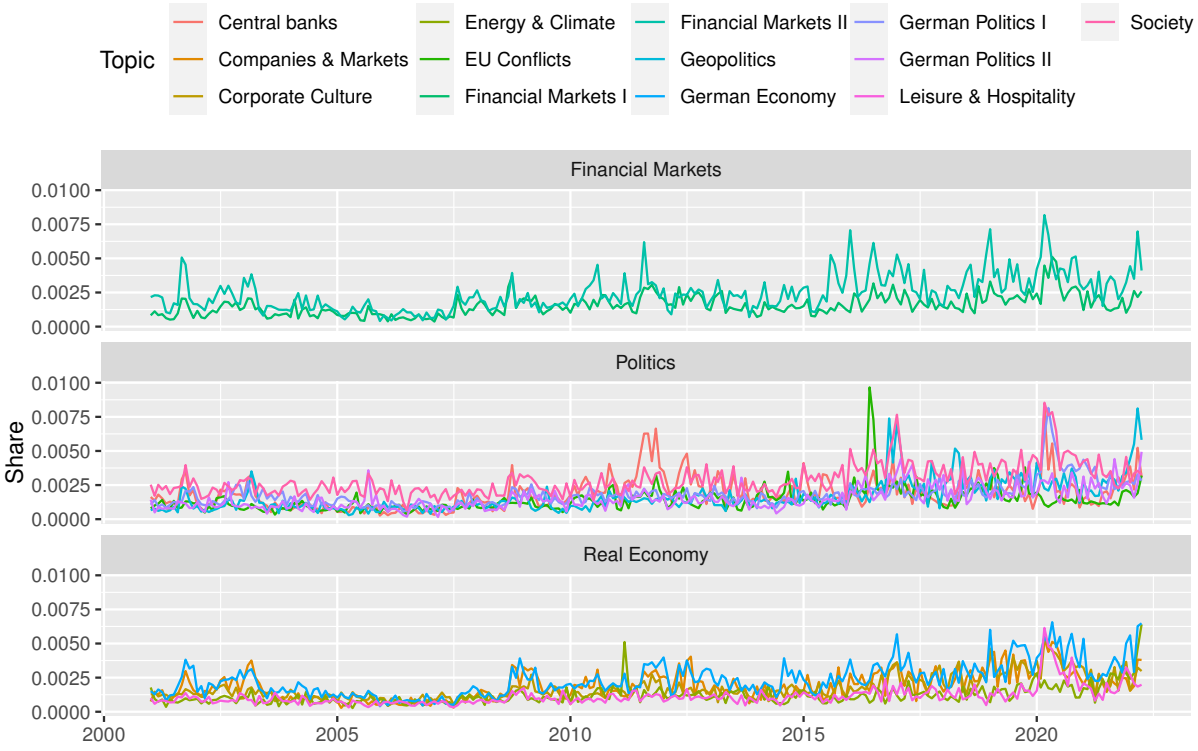


Figure 7: The temporal course of the 13 topics of the UPI split by their association to the three subindices as a monthly share of the total coverage of the newspapers *Die Welt*, *Handelsblatt* and *Süddeutsche Zeitung* from 01/01/2001 until 04/30/2022.

uncertainty. In addition, it may happen that some (long) articles are correctly included in the subcorpus, but also contain thematically inappropriate sections. It is also likely for this case that many assignments from LDA are made to the corresponding topic *Miscellaneous*. By manual coding, we assign the 13 meaningful topics to three subindices *Financial Markets*, *Politics*, and *Real Economy*. This topic is not considered for the calculation of the total UPI and any subindices. Every single index is calculated as the share of the topic measured by the number of words of the total coverage in the complete corpus.

In Figure 7, the indices of the 13 topics of the UPI are shown according to their association with one of the three subindices. By some examples you can see that the method provides plausible results - in an automated way. The financial crisis of 2008 can be identified by an increased share in all three subindices; in *Real Economy* the share remains at a higher level for a longer period of time. The euro crisis of 2010/11 is particularly characterized by a sharp rise of the *Central banks* topic. The clearest peak is caused by the discussions about the United Kingdom's exit (Brexit) from the European Union (EU) in 2016, observed in the *EU Conflicts* topic. The Covid-19 pandemic had the greatest impact on the overall UPI so far. At the beginning of 2020, many topics are rising rapidly; the topics *Leisure & Hospitality (Real Economy)* and *Society (Politics)* are most notable.

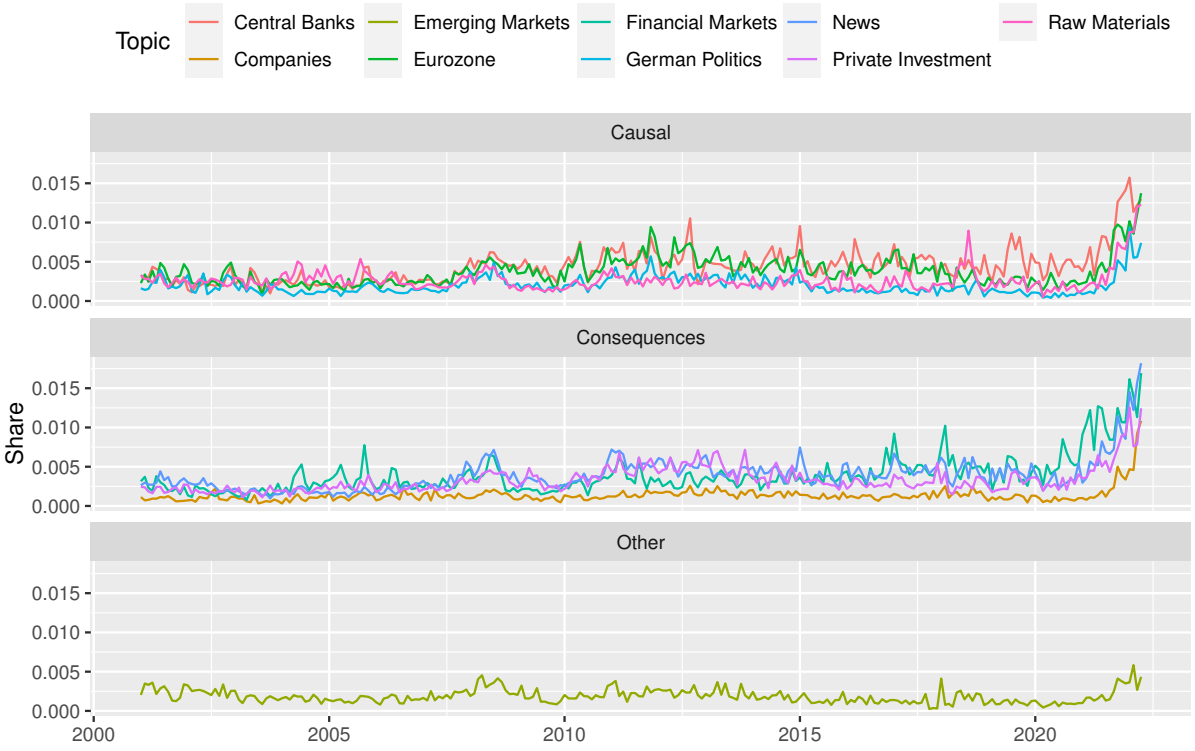


Figure 8: The temporal course of the nine topics of the IPI split by their rough type of relation to inflation as a monthly share of the total coverage of the newspapers *Die Welt*, *Handelsblatt* and *Süddeutsche Zeitung* from 01/01/2001 until 04/30/2022.

In the topic *Geopolitics*, which was previously only triggered by the Brexit, the effect of the war in the Ukraine on the coverage can also clearly be seen. While this topic's share already peaked in February and March, it decreases slightly in April, while the topics involving consequences for Germany, i.e., *German Economy* and *Energy & Climate* rise sharply at the current boundary in April. In general, it can be observed that peaks in the *Financial Markets* subindex are usually short-lived, while the other two subindices remain at a higher level for a longer period of time and slowly decline. This could be because of longer attention cycles due to the transition of attention through the different topics that make up the two subindices.

In addition to the general uncertainty, there is an increased interest in inflation reporting not only among economists, but also among the population as a whole, in particular due to the unexpected post-Covid inflation. Analogously to the UPI, the inflation perception indicator (IPI, Müller et al., 2022b) reflects the intensity of the various facets of inflation reporting given by the share of nine topics (ten modeled topics). The model is updated monthly and the last three months are used as memory, so that changes in the topics are allowed to occur more quickly and temporary narratives in topics are forgotten more quickly than in the thematically broader UPI. In Figure 8, the shares of the nine IPI

topics to the total corpus are shown until the end of April. The subcorpus contains 51 910 articles up to this point. The plausibility of the IPI modeling can also be justified by a few examples. The financial crisis in 2008 is reflected in particular in the four *Consequences* topics. Within the four *Causal* topics, however, the effects on the share of reporting are less strong and less sudden. Already in 2010, high shares become apparent in the *Central banks* topic; in 2011, the *Financial Markets*, *Private Investment* and *News* topics from the *Consequences* topics also show an increase. These increases are likely to be caused by the euro crisis. In 2018, the trade war between China and the United States of America can be seen in a sudden increase in the *Raw Materials (Causal)* topic, and the European Central Bank's (ECB) decision to continue quantitative easing is reflected in an increase in the *Central banks (Causal)* topic over at least three months in 2019. At the current boundary of the modeling, a clear trend can be observed. All nine curves reached their all-time high within the last three months, and further increases are plausible or even likely for some of them. The *Financial Markets (Consequences)* topic already shows spikes at the end of 2020, which are followed by a sharp and sudden increase in the share of reporting in all nine topics by mid-2021 at the latest.

When interpreting the UPI curves, it is important to keep in mind that both, an increase in existing uncertainty may lead to more reporting of it and, vice versa, more attention to uncertainty may lead to just that, as part of a self-fulfilling prophecy. The relationship of IPI to actual inflation should also be analyzed under these constraints.

3.5 Outlook: Dynamic parameter adjustment and temporal topics

In extension to monitoring time series resulting from the model, topics should also be monitored for consistency over time to ensure interpretability. In Rieger et al. (2022b), we monitor similarities of topics to themselves across time points. We also propose a methodology for automated detection of changes in topics. For this, using cosine similarity, we compare the realized similarity of the topic sequence with the similarity between the past and expected subsequent word vectors under semi-stable conditions. If the actual similarity is below the expected stability, the method detects a change. We demonstrate the plausibility of the results by applying the method to a sample dataset of Covid-19 related online news from CNN. Due to the properties of news, we chose a weekly update rhythm and use the texts from the previous week as memory in each chunk. In Figure 6b on page 25, the modeling procedure using RollingLDA is shown schematically.

Based on the results of detected changes, one can think of further approaches to develop extensions of the modeling procedure. A substantial change in a topic suggests that the content has changed significantly or that the previous topic is no longer represented at all.

In such cases, post-processing of the time series of topics becomes useful, so that clearly temporally defined topics within one modeled topic are split into several model topics. This kind of lifetime determination of the topics extends the possibilities of visualization and utilization of the LDA results and thus the interpretability for end users. An advanced further approach is then the implementation of time-constrained topics during the modeling process. This is possible, for example, by model flexibilization towards time-dependent model parameters. Detected changes in topics can then - depending on the type of detected change - e.g., result in the termination of the topic or the creation of a new topic, so that in particular the parameter K, which specifies the number of topics in the model, is not rigid, but may vary time-dependently in an appropriate range around the global value K.

In addition, there are many fields of application of the presented methodology. As a side effect of the digitalization and the fast consumption of news, these seem to result in shorter attention cycles. Lorenz-Spreen et al., 2019 were able to show this using seven different online text datasets as examples. Following this study, the question arises whether narratives in newspapers also tend to have shorter life cycles as time passes. To be able to research this, it is essential to first define the narrative term precisely and reasonably. The four elements of a media frame according to Entman (1993) *a) a problem definition, b) a problem diagnosis, c) a moral judgment, and d) possible remedies* as well as the findings from the work of Matthes and Kohring (2008) and DiMaggio et al. (2013), who propose under which conditions topics from topic models can also be interpreted as frames, should be considered and might be extended to a definition of a narrative concept (cf. Müller et al., 2018).

In the monitoring context, I aim to incorporate temporal topics by integrating a dynamic adjustment of the model parameters - especially the number of topics K - within the modeling. Furthermore, it seems quite realistic and plausible to be able to show with the help of the presented methodology that the narrative cycles in newspapers have become shorter in the last years/decades, i.e., that media coverage has become more fast-paced.

References

Agrawal, Amritanshu, Wei Fu, and Tim Menzies (2018). "What is wrong with topic modeling? And how to fix it using search-based software engineering". In: *Information and Software Technology* 98, pp. 74–88. DOI: 10.1016/j.infsof.2018.02.005.

ALDayel, Abeer and Walid Magdy (2021). "Stance detection on social media: State of the art and trends". In: *Information Processing & Management* 58.4, p. 102597. DOI: 10.1016/j.ipm.2021.102597.

Aletras, Nikolaos and Mark Stevenson (2013). "Evaluating Topic Coherence Using Distributional Semantics". In: *Proceedings of the 10th IWCS-Conference – Long Papers*. ACL, pp. 13–22. URL: https://aclanthology.org/W13-0102.

Aletras, Nikolaos and Mark Stevenson (2014). "Measuring the Similarity between Automatically Generated Topics". In: *Proceedings of the 14th EACL-Conference, Volume 2: Short Papers*. ACL, pp. 22–27. DOI: 10.3115/v1/E14-4005.

Amoualian, Hesam, Marianne Clausel, Eric Gaussier, and Massih-Reza Amini (2016). "Streaming-LDA: A Copula-Based Approach to Modeling Topic Dependencies in Document Streams". In: *Proceedings of the 22nd SIGKDD-Conference*. ACM, pp. 695–704. DOI: 10.1145/2939672.2939781.

Asuncion, Arthur, Max Welling, Padhraic Smyth, and Yee Whye Teh (2009). "On Smoothing and Inference for Topic Models". In: *Proceedings of the 25th UAI-Conference*. AUAI, pp. 27–34. URL: https://dl.acm.org/doi/10.5555/1795114.1795118.

Baker, Scott R., Nicholas Bloom, and Steven J. Davis (2016). "Measuring Economic Policy Uncertainty". In: *The Quarterly Journal of Economics* 131.4, pp. 1593–1636. DOI: 10.1093/qje/qjw024.

Benoit, Kenneth, Michael Laver, and Slava Mikhaylov (2009). "Treating Words as Data with Error: Uncertainty in Text Statements of Policy Positions". In: *American Journal of Political Science* 53.2, pp. 495–513. DOI: 10.1111/j.1540-5907.2009.00383.x.

Bischl, Bernd, Michel Lang, and Patrick Schratz (2020). *parallelMap: Unified Interface to Parallelization Back-Ends*. R package version 1.5.0. URL: https://CRAN.R-project.org/package=parallelMap.

Blei, David M. (2012). "Probabilistic Topic Models". In: *Communications of the ACM* 55.4, pp. 77–84. DOI: 10.1145/2133806.2133826.

Blei, David M. and John D. Lafferty (2006). "Dynamic Topic Models". In: *Proceedings of the 23rd ICML-Conference*. ACM, pp. 113–120. DOI: 10.1145/1143844.1143859.

Blei, David M. and John D. Lafferty (2007). "A Correlated Topic Model of Science". In: *The Annals of Applied Statistics* 1.1, pp. 17–35. DOI: 10.1214/07-AOAS114.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan (2003). "Latent Dirichlet Allocation". In: *Journal of Machine Learning Research* 3, pp. 993–1022. DOI: 10.1162/jmlr.2003.3.4-5.993.

Bose, Avinandan and Soumendu Sundar Mukherjee (2021). "Changepoint Analysis of Topic Proportions in Temporal Text Data". In: *arXiv*. DOI: 10.48550/arXiv.2112.00827.

Chang, Jonathan (2015). *lda: Collapsed Gibbs Sampling Methods for Topic Models*. R package version 1.4.2. URL: https://CRAN.R-project.org/package=lda.

Chang, Jonathan, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei (2009). "Reading Tea Leaves: How Humans Interpret Topic Models". In: *NIPS: Advances in Neural Information Processing Systems*. Vol. 22. Curran Associates Inc., pp. 288–296. URL: https://papers.nips.cc/paper/2009/hash/f92586a25bb3145facd64ab20fd554ff-Abstract.html.

Chien, Jen-Tzung and Meng-Sung Wu (2008). "Adaptive Bayesian Latent Semantic Analysis". In: *IEEE Transactions on Audio, Speech, and Language Processing* 16.1, pp. 198–207. DOI: 10.1109/TASL.2007.909452.

Dabre, Raj, Chenhui Chu, and Anoop Kunchukuttan (2020). "A Survey of Multilingual Neural Machine Translation". In: *ACM Computing Surveys* 53.5. DOI: 10.1145/3406095.

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

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 NAACL-Conference, Volume 1 (Long and Short Papers)*. ACL, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

Dieng, Adji B., Francisco J. R. Ruiz, and David M. Blei (2020). "Topic Modeling in Embedding Spaces". In: *Transactions of the Association for Computational Linguistics* 8, pp. 439–453. DOI: 10.1162/tacl_a_00325.

DiMaggio, Paul, Manish Nag, and David Blei (2013). "Exploiting affinities between topic modeling and the sociological perspective on culture: Application to newspaper coverage of U.S. government arts funding". In: *Poetics* 41.6, pp. 570–606. DOI: 10.1016/j.poetic.2013.08.004.

Ding, Ran, Ramesh Nallapati, and Bing Xiang (2018). "Coherence-Aware Neural Topic Modeling". In: *Proceedings of the 2018 EMNLP-Conference*. ACL, pp. 830–836. DOI: 10.18653/v1/D18-1096.

Doogan, Caitlin and Wray Buntine (2021). "Topic Model or Topic Twaddle? Re-evaluating Semantic Interpretability Measures". In: *Proceedings of the 2021 NAACL-Conference*. ACL, pp. 3824–3848. DOI: 10.18653/v1/2021.naacl-main.300.

Entman, Robert M. (1993). "Framing: Toward Clarification of a Fractured Paradigm". In: *Journal of Communication* 43.4, pp. 51–58. DOI: 10.1111/j.1460-2466.1993.tb01304.x.

Ethayarajh, Kawin and Dan Jurafsky (2020). "Utility is in the Eye of the User: A Critique of NLP Leaderboards". In: *Proceedings of the 2020 EMNLP-Conference*. ACL, pp. 4846–4853. DOI: 10.18653/v1/2020.emnlp-main.393.

Frermann, Lea and Mirella Lapata (2016). "A Bayesian Model of Diachronic Meaning Change". In: *Transactions of the Association of Computational Linguistics* 4, pp. 31–45. DOI: 10.1162/tacl_a_00081.

Gandomi, Amir and Murtaza Haider (2015). "Beyond the hype: Big data concepts, methods, and analytics". In: *International Journal of Information Management* 35.2, pp. 137–144. DOI: 10.1016/j.ijinfomgt.2014.10.007.

Garnitz, Johanna, Robert Lehmann, and Klaus Wohlrabe (2019). "Forecasting GDP all over the world using leading indicators based on comprehensive survey data". In: *Applied Economics* 51.54, pp. 5802–5816. DOI: 10.1080/00036846.2019.1624915.

Gentzkow, Matthew, Bryan Kelly, and Matt Taddy (2019). "Text as Data". In: *Journal of Economic Literature* 57.3, pp. 535–574. DOI: 10.1257/jel.20181020.

Greene, Derek, Derek O'Callaghan, and Pádraig Cunningham (2014). "How Many Topics? Stability Analysis for Topic Models". In: *ECML PKDD: Machine Learning and Knowledge Discovery in Databases*. Vol. 8724. LNCS. Springer, pp. 498–513. DOI: 10.1007/978-3-662-44848-9_32.

Griffiths, Thomas L. and Mark Steyvers (2004). "Finding scientific topics". In: *Proceedings of the National Academy of Sciences* 101.suppl 1, pp. 5228–5235. ISSN: 0027-8424. DOI: 10.1073/pnas.0307752101.

Grimmer, Justin and Brandon M. Stewart (2013). "Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts". In: *Political Analysis* 21.3, pp. 267–297. DOI: 10.1093/pan/mps028.

Hoffman, Matthew, Francis Bach, and David Blei (2010). "Online Learning for Latent Dirichlet Allocation". In: *NIPS: Advances in Neural Information Processing Systems*. Vol. 23. Curran Associates, Inc., pp. 856–864. URL: https://papers.nips.cc/paper/2010/hash/71f6278d140af599e06ad9bf1ba03cb0-Abstract.html.

Hofmann, Thomas (1999). "Probabilistic Latent Semantic Indexing". In: *Proceedings of the 22nd International SIGIR-Conference*. ACM, pp. 50–57. DOI: 10.1145/312624.312649.

Hofmann, Thomas (2001). "Unsupervised Learning by Probabilistic Latent Semantic Analysis". In: *Machine Learning* 42, pp. 177–196. DOI: 10.1023/A:1007617005950.

Hong, Liangjie, Byron Dom, Siva Gurumurthy, and Kostas Tsioutsiouliklis (2011). "A Time-Dependent Topic Model for Multiple Text Streams". In: *Proceedings of the 17th SIGKDD-Conference*. ACM, pp. 832–840. DOI: 10.1145/2020408.2020551.

Hoyle, Alexander, Pranav Goel, Andrew Hian-Cheong, Denis Peskov, Jordan Lee Boyd-Graber, and Philip Resnik (2021). "Is Automated Topic Model Evaluation Broken? The Incoherence of Coherence". In: *NeurIPS: Advances in Neural Information Processing Systems*. URL: https://openreview.net/forum?id=tjdHCnPqoo.

Jankowicz, Nina (2020). *How to Lose the Information War: Russia, Fake News, and the Future of Conflict*. I.B. Tauris. DOI: 10.5040/9781838607715.

Jentsch, Carsten, Enno Mammen, Henrik Müller, Jonas Rieger, and Christof Schötz (2021). "Text mining methods for measuring the coherence of party manifestos for the German federal elections from 1990 to 2021". In: *DoCMA Working Paper* #8. DOI: 10.17877/de290r-22363.

Joseph, Kenneth, Sarah Shugars, Ryan Gallagher, Jon Green, Alexi Quintana Mathé, Zijian An, and David Lazer (2021). "(Mis)alignment Between Stance Expressed in Social Media Data and Public Opinion Surveys". In: *Proceedings of the 2021 EMNLP-Conference*. ACL, pp. 312–324. DOI: 10.18653/v1/2021.emnlp-main.27.

Keane, Nathan, Connie Yee, and Liang Zhou (2015). "Using Topic Modeling and Similarity Thresholds to Detect Events". In: *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*. ACL, pp. 34–42. DOI: 10.3115/v1/W15-0805.

Kim, Taehoon and Jaesik Choi (2015). "Reading documents for bayesian online change point detection". In: *Proceedings of the 2015 EMNLP-Conference*. ACL, pp. 1610–1619. DOI: 10.18653/v1/D15-1184.

Kocmi, Tom, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes (2021). "To Ship or Not to Ship: An Extensive Evaluation of Automatic Metrics for Machine Translation". In: *Proceedings of the Sixth Conference on Machine Translation*. ACL, pp. 478–494.

Koltcov, Sergei, Sergey I. Nikolenko, Olessia Koltsova, Vladimir Filippov, and Svetlana Bodrunova (2016). "Stable Topic Modeling with Local Density Regularization". In: *Internet Science*. Vol. 9934. LNCS. Springer, pp. 176–188. DOI: 10.1007/978-3-319-45982-0_16.

Koppers, Lars, Jonas Rieger, Karin Boczek, and Gerret von Nordheim (2020). *tosca: Tools for Statistical Content Analysis*. R package version 0.2-0. DOI: 10.5281/zenodo.3591068.

Lang, Michel, Bernd Bischl, and Dirk Surmann (2017). "batchtools: Tools for R to work on batch systems". In: *The Journal of Open Source Software* 2.10. DOI: 10.21105/joss.00135.

Lau, Jey Han, David Newman, and Timothy Baldwin (2014). "Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality". In: *Proceedings of the 14th EACL-Conference*. ACL, pp. 530–539. DOI: 10.3115/v1/E14-1056.

Liang, Qiao and Kaibo Wang (2019). "Monitoring of user-generated reviews via a sequential reverse joint sentiment-topic model". In: *Quality and Reliability Engineering International* 35.4, pp. 1180–1199. DOI: 10.1002/qre.2452.

Lorenz-Spreen, Philipp, Bjarke Mørch Mønsted, Philipp Hövel, and Sune Lehmann (2019). "Accelerating dynamics of collective attention". In: *Nature Communications* 10.1759. DOI: 10.1038/s41467-019-09311-w.

Maier, Daniel, A. Waldherr, P. Miltner, G. Wiedemann, A. Niekler, A. Keinert, B. Pfetsch, G. Heyer, U. Reber, T. Häussler, H. Schmid-Petri, and S. Adam (2018). "Applying LDA Topic Modeling in Communication Research: Toward a Valid and Reliable Methodology". In: *Communication Methods and Measures* 12.2-3, pp. 93–118. DOI: 10.1080/19312458.2018.1430754.

Mantyla, Mika V., Maelick Claes, and Umar Farooq (2018). "Measuring LDA Topic Stability from Clusters of Replicated Runs". In: *Proceedings of the 12th ACM/IEEE International ESEM-Symposium*. ACM. ISBN: 9781450358231. DOI: 10.1145/3239235.3267435.

Mathur, Nitika, Timothy Baldwin, and Trevor Cohn (2020). "Tangled up in BLEU: Reevaluating the Evaluation of Automatic Machine Translation Evaluation Metrics". In: *Proceedings of the 58th ACL-Conference*. ACL, pp. 4984–4997. DOI: 10.18653/v1/2020.acl-main.448.

Matthes, Jörg and Matthias Kohring (2008). "The Content Analysis of Media Frames: Toward Improving Reliability and Validity". In: *Journal of Communication* 58.2, pp. 258–279. DOI: 10.1111/j.1460-2466.2008.00384.x.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean (2013). "Distributed Representations of Words and Phrases and their Compositionality". In: *NIPS: Advances in Neural Information Processing Systems*. Vol. 26. Curran Associates, Inc., pp. 3111–3119. URL: https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html.

Mimno, David, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum (2011). "Optimizing Semantic Coherence in Topic Models". In: *Proceedings of the 2011 EMNLP-Conference*. ACL, pp. 262–272. URL: https://dl.acm.org/doi/10.5555/2145432.2145462.

Morstatter, Fred and Huan Liu (2018). "In Search of Coherence and Consensus: Measuring the Interpretability of Statistical Topics". In: *Journal of Machine Learning Research* 18.169, pp. 1–32. URL: http://jmlr.org/papers/v18/17-069.html.

Müller, Henrik, Gerret von Nordheim, Karin Boczek, Lars Koppers, and Jörg Rahnenführer (2018). "Der Wert der Worte – Wie digitale Methoden helfen, Kommunikations- und Wirtschaftswissenschaft zu verknüpfen". In: *Publizistik* 63.4. German. [The value of words – How digital methods help to link communication and economics.], pp. 557–582. DOI: 10.1007/s11616-018-0461-x.

Müller, Henrik, Jonas Rieger, and Nico Hornig (2022a). "Vladimir vs. the Virus – a Tale of two Shocks. An Update on our Uncertainty Perception Indicator (UPI) to April 2022 – a Research Note". In: *DoCMA Working Paper* #11. *Previous versions:* "Riders on the Storm" (Q1 2021), "We're rolling" (Q4 2020), "For the times they are a-changin'" (Q3 2020). DOI: 10.17877/DE290R-22780.

Müller, Henrik, Jonas Rieger, Tobias Schmidt, and Nico Hornig (2022b). "Pressure is high – and rising: The Inflation Perception Indicator (IPI) to 30 April 2022 – a Research Note Analysis". In: *DoCMA Working Paper* #10. *Previous versions:* "A German Inflation Narrative" (02/28/2022). DOI: 10.17877/DE290R-22769.

Newman, David, Edwin V. Bonilla, and Wray Buntine (2011). "Improving Topic Coherence with Regularized Topic Models". In: *NIPS: Advances in Neural Information Processing Systems*. Vol. 24. Curran Associates Inc., pp. 496–504. URL: https://proceedings.neurips.cc/paper/2011/hash/5ef698cd9fe650923ea331c15af3b160-Abstract.html.

Newman, David, Jey Han Lau, Karl Grieser, and Timothy Baldwin (2010). "Automatic Evaluation of Topic Coherence". In: *Proceedings of the 2010 NAACL-Conference*. ACL, pp. 100–108. URL: https://aclanthology.org/N10-1012.

Nguyen, Dat Quoc, Richard Billingsley, Lan Du, and Mark Johnson (2015). "Improving Topic Models with Latent Feature Word Representations". In: *Transactions of the Association for Computational Linguistics*, pp. 299–313. DOI: 10.1162/tacl_a_00140.

Nguyen, Viet-An, Jordan Boyd-Graber, and Philip Resnik (2014). "Sometimes Average is Best: The Importance of Averaging for Prediction using MCMC Inference in Topic Modeling". In: *Proceedings of the 2014 EMNLP-Conference*. ACL, pp. 1752–1757. DOI: 10.3115/v1/D14-1182.

Oshikawa, Ray, Jing Qian, and William Yang Wang (2020). "A Survey on Natural Language Processing for Fake News Detection". In: *Proceedings of the 12th LREC-Conference*. ELRA, pp. 6086–6093. URL: https://aclanthology.org/2020.lrec-1.747.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002). "BLEU: A Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th ACL-Conference*. ACL, pp. 311–318. DOI: 10.3115/1073083.1073135.

Rieger, Jonas (2020). "ldaPrototype: A method in R to get a Prototype of multiple Latent Dirichlet Allocations". In: *Journal of Open Source Software* 5.51, p. 2181. DOI: 10.21105/joss.02181.

Rieger, Jonas (2021). *rollinglda: Construct Consistent Time Series from Textual Data*. R package version 0.1.0. DOI: 10.5281/zenodo.5266717. URL: https://github.com/JonasRieger/rollinglda.

Rieger, Jonas, Carsten Jentsch, and Jörg Rahnenführer (2020a). "Assessing the Uncertainty of the Text Generating Process Using Topic Models". In: *ECML PKDD 2020 Workshops*. Vol. 1323. CCIS. Springer, pp. 385–396. DOI: 10.1007/978-3-030-65965-3_26.

Rieger, Jonas, Carsten Jentsch, and Jörg Rahnenführer (2021). "RollingLDA: An Update Algorithm of Latent Dirichlet Allocation to Construct Consistent Time Series from Textual Data". In: *Findings Proceedings of the 2021 EMNLP-Conference*. ACL, pp. 2337–2347. DOI: 10.18653/v1/2021.findings-emnlp.201.

Rieger, Jonas, Carsten Jentsch, and Jörg Rahnenführer (2022a). "LDAPrototype: A Model Selection Algorithm to Improve Reliability of Latent Dirichlet Allocation". In: *Preprint available at Research Square*. DOI: 10.21203/rs.3.rs-1486359/v1.

Rieger, Jonas, Kai-Robin Lange, Jonathan Flossdorf, and Carsten Jentsch (2022b). "Dynamic change detection in topics based on rolling LDAs". In: *Proceedings of the Text2Story'22 Workshop*. Vol. 3117. CEUR-WS, pp. 5–13. URL: http://ceur-ws.org/Vol-3117/.

Rieger, Jonas, Jörg Rahnenführer, and Carsten Jentsch (2020b). "Improving Latent Dirichlet Allocation: On Reliability of the Novel Method LDAPrototype". In: *Natural Language Processing and Information Systems, NLDB 2020*. Vol. 12089. LNCS. Springer, pp. 118–125. DOI: 10.1007/978-3-030-51310-8_11.

Roberts, Margaret E., Brandon M. Stewart, Dustin Tingley, and Edoardo M. Airoldi (2013). "The Structural Topic Model and Applied Social Science". In: *NIPS-Workshop on Topic Models: Computation, Application, and Evaluation*.

Röder, Michael, Andreas Both, and Alexander Hinneburg (2015). "Exploring the Space of Topic Coherence Measures". In: *Proceedings of the 8th WSDM Conference*. ACM, pp. 399–408. DOI: 10.1145/2684822.2685324.

Rosen-Zvi, Michal, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth (2004). "The Author-Topic Model for Authors and Documents". In: *Proceedings of the 20th UAI-Conference*. AUAI, pp. 487–494. URL: https://dl.acm.org/doi/10.5555/1036843.1036902.

Shilakes, Christopher C. and Julie Tylman (1998). "Move Over Yahoo!; the Enterprise Information Portal Is on Its Way". In: *Enterprise Information Portals*. URL: https://web.archive.org/web/20110724175845/http://ikt.hia.no/perep/eip_ind.pdf.

Stevens, Keith, Philip Kegelmeyer, David Andrzejewski, and David Buttler (2012). "Exploring Topic Coherence over Many Models and Many Topics". In: *Proceedings of the 2012 Joint EMNLP/CoNLL-Conference*. ACL, pp. 952–961. URL: https://aclanthology.org/D12-1087.

Su, Jing, Derek Greene, and Oisin Boydell (2016). "Topic Stability over Noisy Sources". In: *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*. COLING, pp. 85–93. URL: http://aclweb.org/anthology/W16-3913.

Teh, Yee, David Newman, and Max Welling (2006). "A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation". In: *NIPS: Advances in Neural Information Processing Systems*. Vol. 19. MIT Press, pp. 1353–1360. URL: https://proceedings.neurips.cc/paper/2006/hash/532b7cbe070a3579f424988a040752f2-Abstract.html.

Thorsrud, Leif Anders (2020). "Words are the New Numbers: A Newsy Coincident Index of the Business Cycle". In: *Journal of Business & Economic Statistics* 38.2, pp. 393–409. DOI: 10.1080/07350015.2018.1506344.

von Nordheim, Gerret and Jonas Rieger (2020). "Im Zerrspiegel des Populismus – Eine computergestützte Analyse der Verlinkungspraxis von Bundestagsabgeordneten auf Twitter". In: *Publizistik* 65. German. [Distorted by Populism – A computational analysis of German parliamentarians' linking practices on Twitter], pp. 403–424. DOI: 10.1007/s11616-020-00591-7.

von Nordheim, Gerret, Jonas Rieger, and Katharina Kleinen-von Königslöw (2021). "From the Fringes to the Core – An Analysis of Right-Wing Populists' Linking Practices in Seven EU Parliaments and Switzerland". In: *Digital Journalism*, pp. 1–19. DOI: 10.1080/21670811.2021.1970602.

Wallach, Hanna M., Iain Murray, Ruslan Salakhutdinov, and David Mimno (2009). "Evaluation Methods for Topic Models". In: *Proceedings of the 26th ICML Conference*. ACM, pp. 1105–1112. DOI: 10.1145/1553374.1553515.

Wang, Chong, David M. Blei, and David Heckerman (2008). "Continuous Time Dynamic Topic Models". In: *Proceedings of the 24th UAI-Conference*. AUAI, pp. 579–586. URL: https://dl.acm.org/doi/10.5555/3023476.3023545.

Wang, Xuerui and Andrew McCallum (2006). "Topics over Time: A Non-Markov Continuous-Time Model of Topical Trends". In: *Proceedings of the 12th SIGKDD-Conference*. ACM, pp. 424–433. DOI: 10.1145/1150402.1150450.

Wang, Yu, Eugene Agichtein, and Michele Benzi (2012). "TM-LDA: Efficient Online Modeling of Latent Topic Transitions in Social Media". In: *Proceedings of the 18th SIGKDD-Conference*. ACM, pp. 123–131. DOI: 10.1145/2339530.2339552.

Wang, Yunli and Cyril Goutte (2018). "Real-time Change Point Detection using On-line Topic Models". In: *Proceedings of the 27th ACL-Conference*. ACL, pp. 2505–2515. URL: https://www.aclweb.org/anthology/C18-1212.

Wieland, Mareike, Gerret Nordheim, and Katharina Kleinen-von Königslöw (2021). "One Recommender Fits All? An Exploration of User Satisfaction With Text-Based News Recommender Systems". In: *Media and Communication* 9.4, pp. 208–221. DOI: 10.17645/mac.v9i4.4241.

Xing, Linzi, Michael J. Paul, and Giuseppe Carenini (2019). "Evaluating Topic Quality with Posterior Variability". In: *Proceedings of the 2019 Joint EMNLP-IJCNLP-Conference*. ACL, pp. 3471–3477. DOI: 10.18653/v1/D19-1349.

Xu, Guixian, Yueting Meng, Zhan Chen, Xiaoyu Qiu, Changzhi Wang, and Haishen Yao (2019). "Research on Topic Detection and Tracking for Online News Texts". In: *IEEE Access* 7, pp. 58407–58418. DOI: 10.1109/ACCESS.2019.2914097.

Yao, Jianan and Alexander G. Hauptmann (2018). "News Recommendation and Filter Bubble". In: *Proceedings of the CIKM 2018 Workshops*. Vol. 2482. CEUR-WS. URL: http://ceur-ws.org/Vol-2482/.

Zhai, Ke and Jordan Boyd-Graber (2013). "Online Latent Dirichlet Allocation with Infinite Vocabulary". In: *Proceedings of the 30th ICML-Conference*. Proceedings of Machine Learning Research. PMLR, pp. 561–569. URL: http://proceedings.mlr.press/v28/zhai13.html.

Zhao, He, Dinh Phung, Viet Huynh, Yuan Jin, Lan Du, and Wray Buntine (2021). "Topic Modelling Meets Deep Neural Networks: A Survey". In: *arXiv*. DOI: 10.48550/arXiv.2103.00498.

Contributed methodological publications

The following methodological publications are appended as originals or accepted manuscripts. Published articles have been reused with the permission of the copyright holder.

Rieger, Jonas (2020). "ldaPrototype: A method in R to get a Prototype of multiple Latent Dirichlet Allocations". In: *Journal of Open Source Software* 5.51, p. 2181. DOI: 10.21105/joss.02181.

Rieger, Jonas, Carsten Jentsch, and Jörg Rahnenführer (2020a). "Assessing the Uncertainty of the Text Generating Process Using Topic Models". In: *ECML PKDD 2020 Workshops*. Vol. 1323. CCIS. Springer, pp. 385–396. DOI: 10.1007/978-3-030-65965-3_26.

Rieger, Jonas, Carsten Jentsch, and Jörg Rahnenführer (2021). "RollingLDA: An Update Algorithm of Latent Dirichlet Allocation to Construct Consistent Time Series from Textual Data". In: *Findings Proceedings of the 2021 EMNLP-Conference*. ACL, pp. 2337–2347. DOI: 10.18653/v1/2021.findings-emnlp.201.

Rieger, Jonas, Carsten Jentsch, and Jörg Rahnenführer (2022a). "LDAPrototype: A Model Selection Algorithm to Improve Reliability of Latent Dirichlet Allocation". In: *Preprint available at Research Square*. DOI: 10.21203/rs.3.rs-1486359/v1.

Rieger, Jonas, Kai-Robin Lange, Jonathan Flossdorf, and Carsten Jentsch (2022b). "Dynamic change detection in topics based on rolling LDAs". In: *Proceedings of the Text2Story'22 Workshop*. Vol. 3117. CEUR-WS, pp. 5–13. URL: http://ceur-ws.org/Vol-3117/.

Rieger, Jonas, Jörg Rahnenführer, and Carsten Jentsch (2020b). "Improving Latent Dirichlet Allocation: On Reliability of the Novel Method LDAPrototype". In: *Natural Language Processing and Information Systems, NLDB 2020*. Vol. 12089. LNCS. Springer, pp. 118–125. DOI: 10.1007/978-3-030-51310-8_11.

The Journal of Open Source Software

ldaPrototype: A method in R to get a Prototype of multiple Latent Dirichlet Allocations

Jonas Rieger[1]

1 TU Dortmund University

DOI: 10.21105/joss.02181

Software

- Review
- Repository
- Archive

Editor: Karthik Ram

Reviewers:

- @tommyjones
- @bstewart

Submitted: 10 March 2020
Published: 16 July 2020

Summary

Topic Modeling (Blei, 2012) is one of the biggest subjects in the field of text data analysis. Here, the Latent Dirichlet Allocation (Blei, Ng, & Jordan, 2003) takes a special position. A large part of scientific text data analyses are based on this model (LDA). The LDA method has a far-reaching disadvantage. Random initialization and conditional reassignments within the iterative process of the Gibbs sampler (Griffiths & Steyvers, 2004) can result in fundamentally different models when executed several times on the same data and with identical parameter sets. This fact greatly limits the scientific reproducibility.

Up to now, the so-called eye-balling method has been used in practice to select suitable results. From a set of models, subjective decisions are made to select the model that seems to fit the data best or, in the worst case, the result that best supports one's hypothesis is chosen. The latter contradicts basically good scientific practice. A different method of objective and automated selection has also become established. A model from a set of LDAs can be determined optimizing the log-likelihood using the perplexity on held-out data. The `R` (R Core Team, 2020) package `topicmodels` (Grün & Hornik, 2011) provides a workflow for this procedure. As an extension, Nguyen, Boyd-Graber, & Resnik (2014) proposed to average different iterations of the Gibbs sampling procedure to achieve an increase of perplexity. The averaging technique has the weakness, that the user does not get token specific assignments to topics, but only averaged topic counts or proportions per text. In addition, Chang, Boyd-Graber, Gerrish, Wang, & Blei (2009) were able to show that selection mechanisms aiming for optimizing likelihood-based measures do not correspond to the human perception of a well-adapted model of text data. Instead, the authors propose a so-called intruder procedure based on human codings. The corresponding methodology is implemented in the package `tosca` (Koppers, Rieger, Boczek, & von Nordheim, 2019).

The `R` package `ldaPrototype` on the other hand determines a prototypical LDA by automated selection from a set of LDAs. The method improves reliability of findings drawn from LDA results (Rieger, Koppers, Jentsch, & Rahnenführer, 2020), which is achieved following a typical statistical approach. For a given combination of parameters, a number of models is calculated (usually about 100), from which that LDA is determined that is most similar to all other LDAs from a set of models. For this purpose pairwise model similarities are calculated using the S-CLOP measure (**S**imilarity of Multiple Sets by **C**lustering with **Lo**cal **P**runing), which can be determined by a clustering procedure of the individual topic units based on topic similarities of the two LDA results considered. The package offers visualization possibilities for comparisons of LDA models based on the clustering of the associated topics. Furthermore, the package supports the repetition of the modeling procedure of the LDA by a simple calculation of the repeated LDA runs.

In addition to the possibility of local parallel computation by connecting to the package `parallelMap` (Bischl & Lang, 2019), there is the possibility to calculate using batch systems on high performance computing (HPC) clusters by integrating helpful functions from the

The Journal of Open Source Software

package `batchtools` (Lang, Bischl, & Surmann, 2017). This is especially helpful if the text corpora contains several hundred of thousands articles and the sequential calculation of 100 or more LDA runs would extend over several days. The modeling of single LDA runs is done with the help of the computation time optimized `R` package `lda` (Chang, 2015), which implements the calculation in C++ code. In general, the package `ldaPrototype` is based on S3 objects and thus extends the packages `lda` and `tosca` by user-friendly display and processing options. Other `R` packages for estimating LDA are `topicmodels` and `mallet` (Mimno, 2013), whereas `stm` (Roberts, Stewart, & Tingley, 2019) offers a powerful framework for Structural Topic Models and `quanteda` (Benoit et al., 2018) is a popular framework for preprocessing and quantitative analysis of text data.

References

Benoit, K., Watanabe, K., Wang, H., Nulty, P., Obeng, A., Müller, S., & Matsuo, A. (2018). quanteda: An R package for the quantitative analysis of textual data. *Journal of Open Source Software*, *3*(30), 774. doi:10.21105/joss.00774

Bischl, B., & Lang, M. (2019). *parallelMap: Unified Interface to Parallelization Back-Ends*. Retrieved from https://CRAN.R-project.org/package=parallelMap

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

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, *3*, 993–1022. doi:10.1162/jmlr.2003.3.4-5.993

Chang, J. (2015). *lda: Collapsed Gibbs Sampling Methods for Topic Models*. Retrieved from https://CRAN.R-project.org/package=lda

Chang, J., Boyd-Graber, J., Gerrish, S., Wang, C., & Blei, D. M. (2009). Reading Tea Leaves: How Humans Interpret Topic Models. In *Proceedings of the 22nd international conference on neural information processing systems*, NIPS (pp. 288–296). Red Hook, NY, USA: Curran Associates Inc. ISBN: 9781615679119

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, *101*(suppl 1), 5228–5235. doi:10.1073/pnas.0307752101

Grün, B., & Hornik, K. (2011). topicmodels: An R Package for Fitting Topic Models. *Journal of Statistical Software*, *40*(13), 1–30. doi:10.18637/jss.v040.i13

Koppers, L., Rieger, J., Boczek, K., & von Nordheim, G. (2019). *tosca: Tools for Statistical Content Analysis*. doi:10.5281/zenodo.3591068

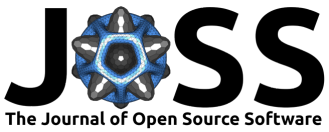
Lang, M., Bischl, B., & Surmann, D. (2017). batchtools: Tools for R to work on batch systems. *The Journal of Open Source Software*, (10). doi:10.21105/joss.00135

Mimno, D. (2013). *mallet: A wrapper around the Java machine learning tool MALLET*. Retrieved from https://CRAN.R-project.org/package=mallet

Nguyen, V.-A., Boyd-Graber, J., & Resnik, P. (2014). Sometimes Average is Best: The Importance of Averaging for Prediction using MCMC Inference in Topic Modeling. In *Proceedings of the 2014 EMNLP-Conference* (pp. 1752–1757). ACL. doi:10.3115/v1/D14-1182

R Core Team. (2020). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Rieger, J., Koppers, L., Jentsch, C., & Rahnenführer, J. (2020). Improving Reliability of Latent Dirichlet Allocation by Assessing Its Stability Using Clustering Techniques on Replicated Runs. Retrieved from https://arxiv.org/abs/2003.04980



Roberts, M. E., Stewart, B. M., & Tingley, D. (2019). stm: An R Package for Structural Topic Models. *Journal of Statistical Software*, *91*(2), 1–40. doi:10.18637/jss.v091.i02

Assessing the Uncertainty of the Text Generating Process using Topic Models

Jonas Rieger[0000−0002−0007−4478], Carsten Jentsch[0000−0001−7824−1697], and Jörg Rahnenführer[0000−0002−8947−440X]

Department of Statistics, TU Dortmund University, 44221 Dortmund, Germany
{rieger,jentsch,rahnenfuehrer}@statistik.tu-dortmund.de

Abstract. Latent Dirichlet Allocation (LDA) is one of the most popular topic models employed for the analysis of large text data. When applied repeatedly to the same text corpus, LDA leads to different results. To address this issue, several methods have been proposed. In this paper, instead of dealing with this methodological source of algorithmic uncertainty, we assess the aleatoric uncertainty of the text generating process itself. For this task, we use a direct LDA-model approach to quantify the uncertainty due to the random process of text generation and propose three different bootstrap approaches to resample texts. These allow to construct uncertainty intervals of topic proportions for single texts as well as for text corpora over time. We discuss the differences of the uncertainty intervals derived from the three bootstrap approaches and the direct approach for single texts and for aggregations of texts. We present the results of an application of the proposed methods to an example corpus consisting of all published articles in a German daily quality newspaper of one full year and investigate the effect of different sample sizes to the uncertainty intervals.

Keywords: Aleatoric Uncertainty · Topic Model · Machine Learning · Stochastic · Text Data

1 Introduction

Modeling unstructured data is a big challenge in the field of machine learning. Due to an increase of volume of unstructured data the need for appropriate analytical methods also increases. Text data covers a large share of unstructured data and is often organized in a collection of texts called corpus.

We consider each text to be a sequence of sentences, where each sentence consists of a sequence of tokens of words. The set of all words are denoted as vocabulary, where a token is given by a word at a specific place in a single text. Observed text data is generally subject to a certain degree of aleatoric uncertainty, since an author uses a slightly different choice of words when writing repeatedly the same text. We provide a mechanism to quantify the uncertainty of the text generation directly and by bootstrap simulation based on topic models. For the bootstrap we distinguish between different implementations of the

procedure, that is, we rely on resampling words, sentences or a combination of both. All approaches are compared regarding uncertainty estimation on an example dataset consisting of all 51 026 articles published in the German quality newspaper Süddeutsche Zeitung in 2018.

1.1 Related Work

In the field of text data analysis it is very common to model a corpus of text data using probabilistic topic models [3]. Latent Dirichlet Allocation [5] is clearly one of the most commonly used topic models and numerous extensions of it have been proposed in the literature that are specialized to certain applications including e.g. the Author-Topic Model [22], Correlated Topics Model [4], or the more generalized Structural Topic Model [21]. However, for illustration we will stick to the classical LDA, but our procedure can be easily extended to other topic models as well.

The modeling procedure of LDA using a Gibbs sampler is stochastic in the sense that it depends on initial topic assignments and reassigns tokens based on conditional distributions. This fact is rarely discussed in applications [1], although several approaches have been proposed to overcome this weakness of algorithmic uncertainty: approaches that optimize perplexity [16] or the semantic coherence of topics [6,14,23] and those, that stabilize the results by initializing the topic assignments reasonably [13,15]. In this paper, for the analysis we will use a new method to select a model of a set of LDAs that is named LDAPrototype [19,20], which will be explained in detail in Sect. 2.1.

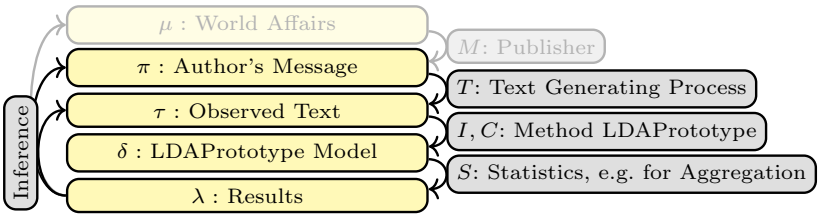


Fig. 1: The stochastic process of text generation: various sources of uncertainty and possible options to do inference on it. Adapted scheme from [2].

When analyzing the evolution of text content over time, it is important to consider various sources of uncertainty in the analysis. Benoit et al. [2] proposed an "Overview of the positions to text to coded data process" visualized in an adapted version in Figure 1. The diagram relates the following components:

- μ, the true unobservable world affairs researched by journalists and effected by strategic decisions by the publisher M, leading to
- π, the underlying intention of a single text's author,
- τ, the observed text generated by a process T, which is often coded by humans based on a coding scheme that someone devised stochastically, denoted with I. In conjunction with the coding process C itself this leads to

- δ, a table of codings, which are modeled using a preliminarily selected modeling procedure S, resulting in
- λ, values for the coded text, which can be used to do inference on
- the observed text τ, the intended message π and the true preference μ.

The uncertainty of C is well known and can be quantified using intercoder reliability measures like Krippendorff's α [12], which offers the opportunity to do inference on the observed text τ. If we can quantify the uncertainty in T or both, T and M, we can do inference on π or μ, respectively. In fact, there are few works (e.g. [2]) that mention the uncertainty in T or M, respectively.

1.2 Contribution

In the following, we take care especially of the aleatoric uncertainty of the text generating process T. Quality newspapers in general claim to reflect the world affairs, which corresponds to μ in Figure 1. Besides the fact that the set of world events is not fully observable, we also cannot infer about them objectively from the published newspaper articles. Publishing companies are influenced by a number of (economic and social) factors to select their contents sensibly and to control their quantity. Moreover, the publishers themselves are not even aware of all the world's incidents. This process is condensed in M and it leads to the unobservable messages of the authors π per event or text, respectively. This is the variable about which we want to make a statement. Therefore, we need to know about the text generating process T that maps the author's message to the observed text τ, the articles of the corpus from the Süddeutsche Zeitung.

We show that three natural variants of bootstrap resampling strategies lead to different degrees of captured uncertainty of the text generating process. In addition, we show that these methods lead to uncertainty intervals for aggregations of articles that are considerably different to the model-based one. Our results suggest that this type of aleatoric uncertainty should be considered in particular for small sample sizes, whereas it becomes negligible for larger sample sizes. In our application, it turns out that the simplest approach of resampling words tends to underestimate the uncertainty of T.

2 Methods

For capturing the aleatoric uncertainty in the text generating process, we make use of topic modeling as the measurement instrument I and the coding process C in the inference scheme of Benoit et al. [2] in Figure 1. That is, in contrast to controlling the stochastic component of the coding process, which is based on manual coding, we need to rule out the algorithmic uncertainty in the modeling process of the topic model.

2.1 Latent Dirichlet Allocation

The topic model we use is a version of the Latent Dirichlet Allocation [5] estimated by a Collapsed Gibbs sampler [10], as a probabilistic topic model that is

widely used in text data analysis. The LDA assumes that there is a topic distribution for every text, and it models them by assigning one topic from the set of topics $\boldsymbol{T} = \{T_1, ..., T_K\}$ to every token in a text, where $K \in \mathbb{N}$ is a user-defined parameter, the number of modeled topics. We denote a text (or document) of a corpus consisting of M texts by $\boldsymbol{D}^{(m)}$, where $N^{(m)}$ is the size of text m, while $\boldsymbol{W} = \{W_1, ..., W_V\}$ is the set of words and $V = |\boldsymbol{W}| \in \mathbb{N}$ the vocabulary size. Then, the topic assignments of every text m are given by $\boldsymbol{T}^{(m)}$:

$$\boldsymbol{D}^{(m)} = \left(W_1^{(m)}, ..., W_{N^{(m)}}^{(m)}\right),\ m = 1, ..., M, \quad W_n^{(m)} \in \boldsymbol{W},\ n = 1, ..., N^{(m)},$$
$$\boldsymbol{T}^{(m)} = \left(T_1^{(m)}, ..., T_{N^{(m)}}^{(m)}\right),\ m = 1, ..., M, \qquad T_n^{(m)} \in \boldsymbol{T},\ n = 1, ..., N^{(m)}.$$

That is, $T_n^{(m)}$ contains the information of topic assignment of the corresponding token $W_n^{(m)}$ in text m. Let $n_k^{(mv)}, k = 1, ..., K, v = 1, ..., V$ denote the number of assignments of word v in text m to topic k. Then, we define the cumulative count of topic k over all words in document m by $n_k^{(m\bullet)}$ and the vectors of topic counts for the $m = 1, ..., M$ texts by $\boldsymbol{t}^{(m)} = (n_1^{(m\bullet)}, ..., n_K^{(m\bullet)})^T$. Using these definitions, the underlying probability model of LDA [10] can be written as

$$W_n^{(m)} \mid T_n^{(m)}, \boldsymbol{\phi}_k \sim \text{ Discrete}(\boldsymbol{\phi}_k), \qquad \boldsymbol{\phi}_k \sim \text{ Dirichlet}(\eta),$$
$$T_n^{(m)} \mid \boldsymbol{\theta}_m \sim \text{ Discrete}(\boldsymbol{\theta}_m), \qquad \boldsymbol{\theta}_m \sim \text{ Dirichlet}(\alpha),$$

where α and η are Dirichlet distribution hyperparameters and must be set by the user. The topic distribution parameters $\boldsymbol{\theta}_m = (\theta_{m,1}, ..., \theta_{m,K})^T \in (0, 1)^K$ can be estimated based on the topic counts $\boldsymbol{t}^{(m)}$. Therefore, Griffiths et al. [10] proposed an estimator including a correction for underestimated topics

$$\hat{\theta}_{m,k} = \frac{n_k^{(m\bullet)} + \alpha}{N^{(m)} + K\alpha}.$$

LDAPrototype Modeling LDAs using a Gibbs sampler is sensitive to the random initialization of topic assignments as mentioned in Sect. 1.1. To control the stochastic nature of LDA we use a recently proposed method to select the "best" model of a set of LDAs. This version of LDA named LDAPrototype [19] leads to an increase of reliability of the conclusions drawn from the prototype model [20]. The increase is obtained by choosing a prototype model as the most central model in the set of all LDA runs, usually from about 100 LDA runs applied to the same dataset. This choice is comparable to the median in the univariate case. The approach is implemented in the R [17] package ldaPrototype [18].

2.2 Text Generating Process

The stochastic component in the process of text generation T in Figure 1 can be quantified model-based or by bootstrap simulation. For the latter case we will use bootstrap resampling of texts.

Model-Based Uncertainty Estimation To estimate the aleatoric uncertainty of the random process of text generation we can use our chosen topic model. That is, we assume the underlying text generating process of the LDA model to represent the truth. We estimate topic proportions by $\hat{\theta}_{m,k}$. For the expected value it holds $\mathrm{E}(\hat{\theta}_{m,k}) = \theta_{m,k}$, if $\hat{\theta}_{m,k}$ is unbiased, that is on average $\hat{\theta}_{m,k}$ reproduces the true unobservable $\theta_{m,k}$. Then, if $\sum_{k=1}^{K} \theta_{m,k} = 1 \ \forall m = 1, ..., M$ and $(\theta_{m,k})_{k=1,...,K}$ are the true, but unobservable topic proportions of the intended message of text m, we obtain from the multinomial distribution

$$\mathrm{E}\left(n_k^{(m\bullet)}\right) = N^{(m)}\hat{\theta}_{m,k} \quad \text{and} \quad \mathrm{Var}\left(n_k^{(m\bullet)}\right) = N^{(m)}\hat{\theta}_{m,k}\left(1-\hat{\theta}_{m,k}\right),$$

$$\text{where } n_k^{(m\bullet)} \sim \text{Binomial}\left(N^{(m)}, \theta_{m,k}\right) \quad \text{and} \quad \boldsymbol{t}^{(m)} \sim \text{Multinomial}\left(N^{(m)}, \boldsymbol{\theta}_m\right).$$

The binomial distribution offers the possibility to calculate an approximate confidence interval for $\theta_{m,k}$ based on the normal distribution [24] given by

$$\mathrm{Var}\left(\hat{\theta}_{m,k}\right) = \frac{\hat{\theta}_{m,k}\left(1-\hat{\theta}_{m,k}\right)}{N^{(m)}} \Rightarrow \mathrm{CI}_{\theta_{m,k}} = \left[\hat{\theta}_{m,k} \pm z_{1-\alpha/2}\sqrt{\frac{\hat{\theta}_{m,k}(1-\hat{\theta}_{m,k})}{N^{(m)}}}\right],$$

where $z_{1-\alpha/2}$ denotes the $(1-\alpha/2)$-quantile of the standard normal distribution. This enables us to quantify the uncertainty of the text generating process of a single text in our corpus based on the chosen topic model LDA. The requirement of a large sample size for an adequate construction of approximate confidence intervals will be not always satisfied for individual texts, but certainly will apply to a (large) number of aggregated texts. Furthermore, it is often of interest to consider those aggregations of texts, especially stratified by time intervals, as for example daily newspaper editions. In Sect. 3, we consider daily newspaper editions of the Süddeutsche Zeitung. However, due to the fact that generally $\mathrm{Cov}(n_k^{(m_1\bullet)}, n_k^{(m_2\bullet)}) \neq 0$ holds for two selected texts $m_1, m_2 \in \{1, ..., M\}$, the analytic derivation of $\mathrm{Var}(n_k^{(m_1\bullet)} + n_k^{(m_2\bullet)})$ is complicated. In this case, the bootstrap approach provides a remedy.

Bootstrap-Based Uncertainty Estimation The seminal idea of the bootstrap [7,8] is that a random sample x relates to its population in the same manner as a random sample x^* drawn independently with replacement from x relates to the random sample x itself. We will use the bootstrap idea to estimate the uncertainty of the text generating process using resampled texts. There are several natural ways how to bootstrap texts. We will propose and investigate three versions:

1. **BW**ord: The text is understood as one bag of words and these words are bootstrapped. The BWord approach corresponds to the model-based assessment and is equivalent to it for a large number of bootstrap replications.
2. **BS**entence: The text is split into sentences and these sentences are put in a imaginary bag of sentences to create new texts generated by resampled sentences. This approach takes more natural structure of the texts into account.

3. **BS**entence**W**ord: This approach is a kind of intermediate of both. At first the BS approach is executed, that is, sentences are resampled. Afterwards, every resampled sentence is resampled with respect to its words. The second step corresponds to a BW approach for sentences.

In the present case we replicate every text 25 000 times using each of the three introduced methods. For aggregation statistics we resample $R = 100\,000$ combinations based on each set of resampled texts. The words in each bootstrapped text are assigned to the corresponding topics from the LDAPrototype model according to the tokens of the original texts. To save computation time, if a text contains less than ten words, in BW we determine all possible combinations of words. The number of possible combinations is given by $\binom{2n-1}{n-1}$, where n denotes the number of individuals. That is, for $n = 9$ we have 24 310 combinations and for $n = 5$ there are only 126 possibilities to combine these five words with replacement. We proceed analogously for the BS approach; for BSW, only if the according text contains one single sentence consisting of less than ten words. Then, we build bootstrap intervals for θ reaching from the 0.025%- to the 0.975%-quantile of $\hat{\theta}_r, r = 1, ..., R$.

As we investigate a daily newspaper, we also calculate daily aggregation statistics using a rolling window approach for 7, 15 and 29 days. That is, for every single day and topic we calculate the mean of the daily count of topic assignments for $\pm 3, \pm 7$ or ± 14 days. The estimator is denoted by $\hat{p}$ and is defined as the simple proportion of topic assignments for each topic at the given day. We determine bootstrap intervals for the real topic proportion p in the style of those for θ. In addition, we measure the relative standard deviation of $\hat{p}$ by the coefficient of variation defined as the ratio of empirical standard deviation and the sample mean $\bar{\hat{p}}$ of the $\hat{p}_r, r = 1, ..., R$, which is given by

$$\mathrm{CV}(\hat{p}) = \frac{1}{\bar{\hat{p}}} \sqrt{\frac{1}{R-1} \sum_{r=1}^{R} \left(\hat{p}_r - \bar{\hat{p}}\right)^2}.$$

This allows us to make uncertainty statements not only in dependence of single topics' proportions but more general depending on the level of $\hat{p}$.

3 Analysis

For the following analysis we refer to a corpus of newspaper articles. We consider the complete set of texts published in the daily German quality newspaper Süddeutsche Zeitung in 2018. It consists of 51 026 articles. We perform classical preprocessing steps for text corpora as removal of numbers, punctuation and distinct German stopwords. In addition, all words that occur ten times or less are deleted from the corpus. The corpus results to consist of $M = 48\,753$ non-empty texts with a vocabulary size of $V = 76\,499$ words. Preprocessing is done with the R packages tosca [11] and tm [9].

Table 1 gives an overview of the resulting number of words or sentences per text and the number of words per sentence. We can conclude that after

Table 1: Statistics of the number of words per text, sentences per text and words per sentence in the preprocessed corpus of Süddeutsche Zeitung in 2018.

	Min	25%	Median	Mean	75%	Max
Words per Text	2	59	164	217	321	2 220
Sentences per Text	1	7	20	28	37	752
Words per Sentence	1	4	7	8	10	171

preprocessing there is a large proportion of rather short texts of less than 300 words, while there is a small number of clearly longer texts with up to 2 220 words. We observe around 28 sentences per text and 8 words per sentence in mean. Mainly due to enumerations of sports results and similar, there is a single text containing 752 sentences and there are 17 sentences with more than 100 words.

3.1 Study Design

We analyze the presented corpus using the proposed methods to quantify the aleatoric uncertainty resulting from the stochastic text generating process T regarding to Figure 1. For this purpose, we assign each word of a text to a topic using the LDAPrototype approach. This corresponds to the coding scheme I (selection of the model) and coding process C (modeling procedure itself). Based on the resulting model of the prototype method δ, the calculation of estimators for topic proportions per document and aggregation statistics lead to the final results λ. These are used to draw conclusions about the observed text τ as well as about the author's message π. In addition, we once generate stratified subcorpora containing 10, 20, 30 or 50% of the articles per day.

Due to the high combinatorial possibility of parameter selection, some parameters have to be chosen arbitrarily but sensibly. We select $K = 50$ and $\alpha = \eta = 1/K = 1/50$ as parameters for the LDA which implies that we assume 50 topics to be present in the corpus. The mixture parameters α and η are selected rather small to create relatively disjoint topics and to meet the intuitive assumption of few but dominant topics per article. In addition, the Gibbs sampler is supposed to iterate 200 times to the final LDA result. The LDAPrototype parameters are all taken from the default setting [18]. Data and scripts can be retrieved from the GitHub repository https://github.com/JonasRieger/edml2020.

3.2 Results

We analyze the corpus in dependence of the different bootstrap approaches and sample sizes. The topic assignment for a specific token is considered to be constant determined by the LDAPrototype to eliminate the algorithmic uncertainty. Hence, the sample size does not effect the estimation for a single document.

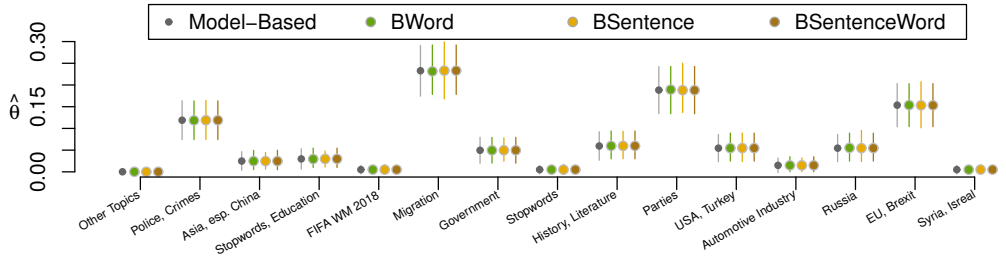


Fig. 2: Estimated topic proportions and corresponding confidence intervals for the article "Wochenchronik vom 30. Juni bis 6. Juli 2018" (weekly chronicle) in Süddeutsche Zeitung published on 7th of July in 2018.

For the analysis of topic proportions on a document level $\hat{\theta}$, we select one article at random. Figure 2 shows the estimation of topic proportions of this random article, which is about a summary of the previous week from the department of politics. The article deals with the topics *Migration*, *Parties*, *EU & Brexit* and *Police & Crimes* at most. This is plausible because the text is mainly about the so-called asylum package that the government parties in Germany adopted on 5th of July. Figure 2 shows that all three bootstrap approaches overall match the model-based confidence interval. Only the intervals resulting from the BS approach differ for some topic proportions slightly from the model-based interval. A reason could be that a certain sentence often consists of one dominating topic, which results in wider intervals for those topics and smaller intervals for topics that are spread over many sentences. The latter is true for the topic *Stopwords & Education*, which appears in almost every sentence because of its composition of stopwords, so that the corresponding uncertainty is rather low. Despite the removal of stopwords there are still stopword topics where less distinct stopwords can be found that were not excluded from the outset.

In the following, we focus on the topic *Migration* for an exemplary analysis of the history of media coverage for a single topic. The proportion of the selected

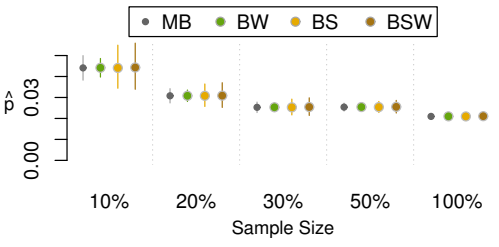


Fig. 3: Estimated topic proportions and corresponding confidence intervals for the topic *Migration* on 7th of July in 2018 for different sample sizes of articles.

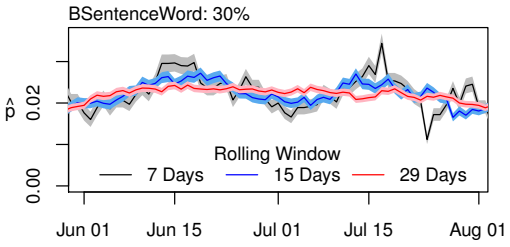


Fig. 4: Smoothed estimated topic proportions and corresponding confidence intervals for the topic *Migration* using the BSW approach and a sample size of 30% from 1st of June to 31st of July in 2018.

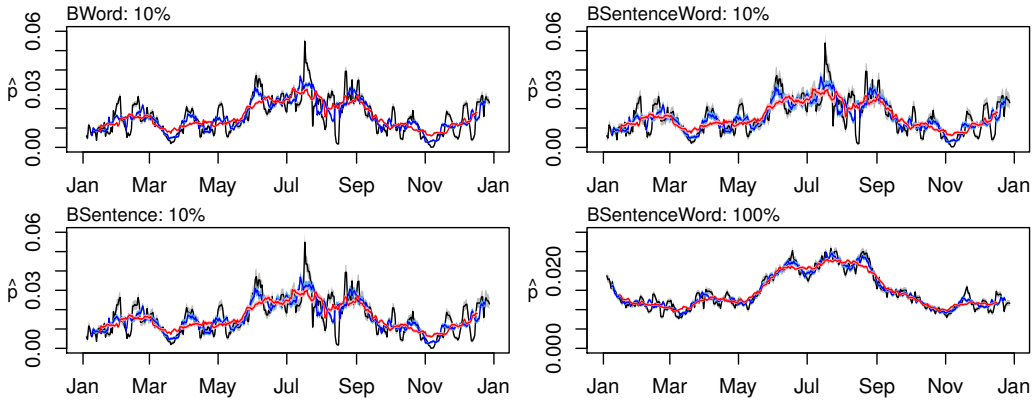


Fig. 5: Smoothed estimated topic proportions and corresponding confidence intervals for the topic *Migration* and for different bootstrap approaches and sample sizes in 2018 using a rolling window over 7 (black), 15 (blue) and 29 days (red).

topic $\hat{p}$ on 7th of July in 2018, corresponding confidence intervals of the bootstrap approaches and the model-based interval in dependence of the sample size can be seen in Figure 3. The LDA-based interval differs from its bootstrap counterpart BW because the implicit assumption that the covariance of two texts equals zero is not fulfilled here (see 2.2). Instead, the uncertainty balances itself out over a large number of articles. The BS intervals are the second widest, while the BSW approach adds some additional uncertainty to it leading to slightly larger intervals. There are 229 articles published at the 7th of July, that is, the sample sizes of 10, 20, 30 and 50% correspond to 22, 45, 68 and 114 articles, respectively. Apparently, a higher number of articles is able to balance itself better, the monotony of the drop in level results from the monotonous selection of articles from the complete data set.

It is common to consider visualizations of topic proportions over time in topic model analysis. Daily topic counts or proportions fluctuate frequently. To overcome this issue, we make use of the rolling window method with window widths of 7, 15 and 29. Figure 5 displays a selection of combinations of those smoothed curves for the topic *Migration*. It is clear that the larger the window size, the smoother the associated curves become and the lower the sample size is, the wider are the intervals that characterize the uncertainty of the text generating process. The topic proportions of the BS and BSW approaches are subject to greater uncertainty than BW. This matches the findings regarding Figure 3.

Figure 4 makes the uncertainty of the topic proportion even more visible. It zooms into the months June and July in 2018 and considers 30% of the articles for the BSW approach. It shows that larger window widths lead to narrower intervals, which can be explained by the better balance through multiple values.

To generalize the results in a proper way, Figure 6 visualizes the standard deviations of topic proportions in dependence of the topic proportion itself. It compares the resampling approaches BW and BSW (results for BS and BSW are similar) and the five sample sizes. For BW the uncertainty is easier to estimate,

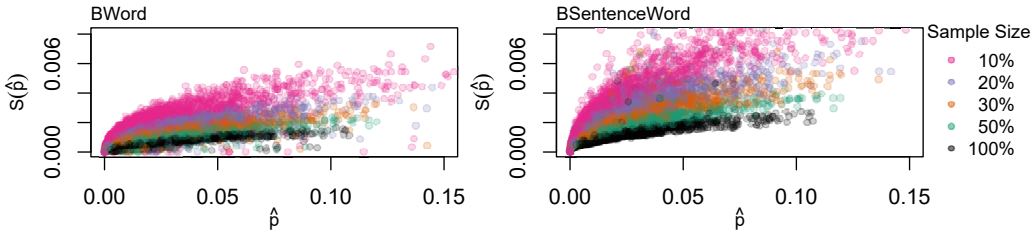


Fig. 6: Coefficient of variation of the daily topic proportion for all topics at all days in 2018, for the BW and BSW approach and for different sample sizes.

the points are not as scattered as for BSW. The estimated uncertainty in dependence of $\hat{p}$ is for BW less pronounced. For example, for $\hat{p} = 0.1$ one should add about 0.001 in uncertainty, while BSW suggest almost 0.002. For smaller sample sizes the effect increases (10%: BW ≈ 0.004, BS ≈ 0.006, BSW > 0.007).

4 Discussion

We have found that for a single article the aleatoric uncertainty of the text generating process can be well quantified by the presented LDA-based confidence interval. The estimator $\hat{\theta}_{m,k}$ is not unbiased, but it shifts the estimator for each topic towards K^{-1} for small sample sizes to represent the usually symmetric chosen a-priori distribution. This robustness property leads to slightly smaller confidence intervals, that matches the more computational-intensive bootstrap intervals to a sufficient extent for all considered articles.

However this intuitive way of calculating confidence intervals is not suitable for aggregated texts as implicitly the covariances $\mathrm{Cov}(n_k^{(m_1\bullet)}, n_k^{(m_2\bullet)})$ are assumed to be zero for the calculation of the LDA-based confidence intervals. Obviously, this is not true in general, leading to intervals that are too small and do not represent appropriately the aleatoric uncertainty of the text generating process for aggregated texts.

For a large number of texts the uncertainty regarding aggregated values becomes negligibly small. On the other hand, especially for small sample sizes, the aleatoric uncertainty of the text generating process should be considered. The three bootstrap methods presented are suitable for this purpose, whereby BW ignores potential dependencies and thus seems to slightly underestimate the uncertainty and BS and BSW are usually very similar. The more intuitive approach of the two seems to be BSW. One can well imagine that authors make use of certain sentence bodies, but these sentences vary slightly. This corresponds exactly to the BSentenceWord approach, which we therefore recommend for assessing the aleatoric uncertainty of the text generating process for aggregated values in small sample size scenarios.

Acknowledgments The present study is part of a project of the Dortmund Center for data-based Media Analysis (DoCMA). In addition, the authors grate-

fully acknowledge the computing time provided on the Linux HPC cluster at Technical University Dortmund (LiDO3), partially funded in the course of the Large-Scale Equipment Initiative by the German Research Foundation (DFG) as project 271512359.

References

1. Agrawal, A., Fu, W., Menzies, T.: What is wrong with topic modeling? And how to fix it using search-based software engineering. Information and Software Technology **98**, 74–88 (2018). https://doi.org/10.1016/j.infsof.2018.02.005
2. Benoit, K., Laver, M., Mikhaylov, S.: Treating words as data with error: Uncertainty in text statements of policy positions. American Journal of Political Science **53**(2), 495–513 (2009). https://doi.org/10.1111/j.1540-5907.2009.00383.x
3. Blei, D.M.: Probabilistic Topic Models. Communications of the ACM **55**(4), 77–84 (2012). https://doi.org/10.1145/2133806.2133826
4. Blei, D.M., Lafferty, J.D.: A Correlated Topic Model of Science. The Annals of Applied Statistics **1**(1), 17–35 (2007). https://doi.org/10.1214/07-AOAS114
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Learning Research **3**, 993–1022 (2003). https://doi.org/10.1162/jmlr.2003.3.4-5.993
6. Chang, J., Boyd-Graber, J., Gerrish, S., Wang, C., Blei, D.M.: Reading Tea Leaves: How Humans Interpret Topic Models. In: Proceedings of the 22nd International NIPS-Conference. pp. 288–296. Curran Associates Inc. (2009), https://dl.acm.org/doi/10.5555/2984093.2984126
7. Efron, B.: Bootstrap methods: Another look at the jackknife. The Annals of Statistics **7**(1), 1–26 (1979), http://www.jstor.org/stable/10.2307/2958830
8. Efron, B., Tibshirani, R.: An Introduction to the Bootstrap. Chapman & Hall/CRC, New York (1994). https://doi.org/10.1201/9780429246593
9. Feinerer, I., Hornik, K., Meyer, D.: Text Mining Infrastructure in R. Journal of Statistical Software **25**(5), 1–54 (2008). https://doi.org/10.18637/jss.v025.i05
10. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proceedings of the National Academy of Sciences **101**(suppl 1), 5228–5235 (2004). https://doi.org/10.1073/pnas.0307752101
11. Koppers, L., Rieger, J., Boczek, K., von Nordheim, G.: tosca: Tools for Statistical Content Analysis (2019). https://doi.org/10.5281/zenodo.3591068, R package version 0.1-5
12. Krippendorff, K.: Content Analysis: An Introduction to Its Methodology. Sage Publications, 3 edn. (2013)
13. Maier, D., Waldherr, A., Miltner, P., Wiedemann, G., Niekler, A., Keinert, A., Pfetsch, B., Heyer, G., Reber, U., Häussler, T., Schmid-Petri, H., Adam, S.: Applying LDA Topic Modeling in Communication Research: Toward a Valid and Reliable Methodology. Communication Methods and Measures **12**(2-3), 93–118 (2018). https://doi.org/10.1080/19312458.2018.1430754
14. Mimno, D., Wallach, H.M., Talley, E., Leenders, M., McCallum, A.: Optimizing Semantic Coherence in Topic Models. In: Proceedings of the 2011 EMNLP-Conference. pp. 262–272. ACL (2011), https://dl.acm.org/doi/10.5555/2145432.2145462
15. Newman, D., Bonilla, E.V., Buntine, W.: Improving Topic Coherence with Regularized Topic Models. In: Proceedings of the 24th International NIPS-Conference. pp.

Reproduced with permission from Springer Nature

496–504. Curran Associates Inc. (2011), https://dl.acm.org/doi/10.5555/2986459.2986515

16. Nguyen, V.A., Boyd-Graber, J., Resnik, P.: Sometimes Average is Best: The Importance of Averaging for Prediction using MCMC Inference in Topic Modeling. In: Proceedings of the 2014 EMNLP-Conference. pp. 1752–1757. ACL (2014). https://doi.org/10.3115/v1/D14-1182
17. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2019), http://www.R-project.org/
18. Rieger, J.: ldaPrototype: A method in R to get a Prototype of multiple Latent Dirichlet Allocations. Journal of Open Source Software **5**(51), 2181 (2020). https://doi.org/10.21105/joss.02181
19. Rieger, J., Koppers, L., Jentsch, C., Rahnenführer, J.: Improving Reliability of Latent Dirichlet Allocation by Assessing Its Stability Using Clustering Techniques on Replicated Runs (2020), https://arxiv.org/abs/2003.04980
20. Rieger, J., Rahnenführer, J., Jentsch, C.: Improving Latent Dirichlet Allocation: On Reliability of the Novel Method LDAPrototype. In: Proceedings of the 25th International NLDB-Conference. LNCS, vol. 12089, pp. 118–125. Springer (2020). https://doi.org/10.1007/978-3-030-51310-8_11
21. Roberts, M.E., Stewart, B.M., Tingley, D., Airoldi, E.M.: The Structural Topic Model and Applied Social Science. In: NIPS-Workshop on Topic Models: Computation, Application, and Evaluation (2013)
22. Rosen-Zvi, M., Griffiths, T., Steyvers, M., Smyth, P.: The Author-Topic Model for Authors and Documents. In: Proceedings of the 20th UAI-Conference. pp. 487–494. AUAI Press (2004), https://dl.acm.org/doi/10.5555/1036843.1036902
23. Stevens, K., Kegelmeyer, P., Andrzejewski, D., Buttler, D.: Exploring Topic Coherence over Many Models and Many Topics. In: Proceedings of the 2012 Joint EMNLP/CoNLL-Conference. pp. 952–961. ACL (2012), https://dl.acm.org/doi/10.5555/2390948.2391052
24. Wald, A.: Tests of statistical hypotheses concerning several parameters when the number of observations is large. Transactions of the American Mathematical Society **54**, 426–482 (1943). https://doi.org/10.1090/S0002-9947-1943-0012401-3

RollingLDA: An Update Algorithm of Latent Dirichlet Allocation to Construct Consistent Time Series from Textual Data

Jonas Rieger and **Carsten Jentsch** and **Jörg Rahnenführer**
Department of Statistics, TU Dortmund University, 44221 Dortmund, Germany
{rieger, jentsch, rahnenfuehrer} @statistik.tu-dortmund.de

Abstract

We propose a rolling version of the Latent Dirichlet Allocation, called RollingLDA. By a sequential approach, it enables the construction of LDA-based time series of topics that are consistent with previous states of LDA models. After an initial modeling, updates can be computed efficiently, allowing for real-time monitoring and detection of events or structural breaks. For this purpose, we propose suitable similarity measures for topics and provide simulation evidence of superiority over other commonly used approaches. The adequacy of the resulting method is illustrated by an application to an example corpus. In particular, we compute the similarity of sequentially obtained topic and word distributions over consecutive time periods. For a representative example corpus consisting of The New York Times articles from 1980 to 2020, we analyze the effect of several tuning parameter choices and we run the RollingLDA method on the full dataset of approximately 4 million articles to demonstrate its feasibility.

1 Introduction

Text data is increasingly used in contexts where structured data is either not available at all or only available with much delay. Hence, text data is often used for the timely detection of events or structural breaks in the context of monitoring over time. This requires first an appropriate modeling methodology and second a suitable analysis methodology. Our new sequential method is based on the well-known and popular model Latent Dirichlet Allocation (LDA, Blei et al., 2003), while assuring that by adding new data the allocations of previously modeled documents do not change. Thus, time series based on the new model are consistent with previous states. We propose the RollingLDA method for modeling consistent and reliable time series on textual data such as topic frequencies on news data. The method uses for each update of new sequential data a previously determined set of documents as a memory. Thus, the method acts like a backward-looking rolling window. In comparison to a lot of existing methods, the presented method does not require recalculation of the whole model when adding new data, which makes it computationally more efficient.

1.1 Related Work

For the selection of suitable tuning parameters, similarity measures for topics are needed. In numerous studies, no clear superiority of one specific measure could be found. Aletras and Stevenson (2014) found out that in most cases the similarity measure using Jensen-Shannon divergence (Lin, 1991) performs as the best similarity measure based on word distributions considering correlation with human judgments. However, they found out that in some cases a Jaccard coefficient (Jaccard, 1912) is able to realize higher correlations to human judgments than other common similarity measures. In accordance, Kim and Oh (2011) showed that Jaccard coefficients perform on par with Jensen-Shannon similarity and outperform a number of other popular similarity measures like cosine similarity, which is commonly used to measure topic similarities (Maier et al., 2018). All of these studies primarily consider similarities of different topics to each other, rather than the similarity of one topic to itself at different points in time.

In contrast, Keane et al. (2015) used cosine similarity for identifying topics characterized by events in daily LDA models. They mention the symmetric Kullback-Leibler divergence (Kullback and Leibler, 1951), that is, the Jensen-Shannon divergence, as a good alternative for computing similarities. The latter is also used by Xu et al. (2019) for studying the evolution of topics in news data. Their study suggests that LDA is a good method for this type of detecting structural breaks in topics. Wang and Goutte (2018) also used LDA models and compare

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 2337–2347
November 7–11, 2021.

cosine similarity and Jensen-Shannon similarity with different change point algorithms on a self-annotated corpus. They found out that online LDA (Zhai and Boyd-Graber, 2013) performs on par with standard LDA for this task. Since no evidence for the consistent superiority of any of the similarity measures could be shown in the available studies, we use and compare different similarity measures for self-similarity of topics.

The calculation of topic similarities should be based on a reliable topic model. For modeling temporal text data, there is the Topics over Time model by Wang and McCallum (2006) or the Dynamic Topic Models by Blei and Lafferty (2006), which was also extended to Continuous Time Dynamic Topic Models by Wang et al. (2008). These methods model the collection of all documents together, so that for new data a recalculation of the whole model is necessary. Besides the computational demand, this may also change previous results depending on how much future text data is added. Hoffman et al. (2010) extended the classical LDA to an online approach, but focused on batches of documents with fixed size rather than time-stamped documents. In addition, Temporal LDA (Wang et al., 2012) is an approach for modeling text streams with LDA using transition matrices. The model is mainly specialized for social media posts, as it assumes streamed texts to be written by the same set of authors. Amoualian et al. (2016) proposed a method called Streaming-LDA. They model dependencies between consecutive documents based on Dirichlet distributions or copula based.

1.2 Contribution

We present a model that is updated when new data is received in a way that ensures consistent time series without the need of recalculation. We combine this update algorithm with classical LDA. To reduce the dependence of LDA results from the initial randomization we use LDAPrototype (Rieger et al., 2020). Another approach would be to average multiple Gibbs iterations (Nguyen et al., 2014). However, as the concrete assignments are lost due to averaging, their approach is not suitable for the RollingLDA method. We do not select a reliable model using likelihood-based measures, e.g., using the package topicmodels (Grün and Hornik, 2011) because Chang et al. (2009) were able to show that these measures are negatively correlated with human perception of good models. An alternative to LDAPrototype for a reliable selection criterion could also be defined based on topic's semantic coherence (Mimno et al., 2011; Stevens et al., 2012).

Our model takes a slightly different approach than the ones mentioned in Sect. 1.1. It considers the set of articles split into intervals or chunks based on its time stamp rather than a real stream. The method focuses on the possibility of evolving topics and the simultaneous monitoring of these changes in a real world scenario of updating an existing LDA model with newly releasing documents. In addition to the proposal of our novel method RollingLDA, we also compare six commonly used similarity measures for topics with respect to their suitability for event detection within topics. Furthermore, these measures can be used as criteria for an individual appropriate choice of the memory parameter in the RollingLDA method.

2 Methodological Framework

The RollingLDA method we propose is based on the classical LDA (Blei et al., 2003) estimated by a collapsed Gibbs sampler (Griffiths and Steyvers, 2004) and we combine it with the method LDAPrototype (Rieger et al., 2020), which selects the most reliable LDA from a set of models.

2.1 Latent Dirichlet Allocation

The classical LDA assumes distributions of latent topics for each text. If K denotes the total number of modeled topics, the set of topics is given by $\boldsymbol{T} = \{T_1, \dots, T_K\}$. We define $W_n^{(m)}$ as a single token at position n in text m. The set of possible tokens is given by the vocabulary $\boldsymbol{W} = \{W_1, \dots, W_V\}$ with $V = |\boldsymbol{W}|$, the vocabulary size. Then, let

$$\boldsymbol{D}^{(m)} = \left(W_1^{(m)}, \dots, W_{N^{(m)}}^{(m)}\right),$$

be text (or document) $m = 1, \dots, M$, of a corpus consisting of M texts. Each text in turn consists of $N^{(m)}$ word tokens $W_n^{(m)} \in \boldsymbol{W}$, $n = 1, \dots, N^{(m)}$. Topics are referred to as $T_n^{(m)} \in \boldsymbol{T}$ for the topic assignment of token $W_n^{(m)}$. Then, analogously the topic assignments of every text m are given by

$$\boldsymbol{T}^{(m)} = \left(T_1^{(m)}, \dots, T_{N^{(m)}}^{(m)}\right).$$

When $n_k^{(mv)}, k = 1, \dots, K, v = 1, \dots, V$ describes the number of assignments of word v in

text m to topic k, we can define the cumulative count of word v in topic k over all documents by $n_k^{(\bullet v)}$ and, analogously, the cumulative count of topic k over all words in document m by $n_k^{(m\bullet)}$, while $n_k^{(\bullet\bullet)}$ indicates the total count of assignments to topic k.

Using these definitions, the underlying probability model (Griffiths and Steyvers, 2004) can be written as

$$\begin{aligned} W_n^{(m)} \mid T_n^{(m)}, \boldsymbol{\phi}_k &\sim \text{Discrete}(\boldsymbol{\phi}_k), \\ \boldsymbol{\phi}_k &\sim \text{Dirichlet}(\eta), \\ T_n^{(m)} \mid \boldsymbol{\theta}_m &\sim \text{Discrete}(\boldsymbol{\theta}_m), \\ \boldsymbol{\theta}_m &\sim \text{Dirichlet}(\alpha). \end{aligned}$$

For a given parameter set $\{K, \alpha, \eta\}$, LDA assigns one of the K topics to each token. Here K denotes the number of topics and α, η are parameters of a Dirichlet distribution defining the type of mixture of topics in every text and the type of mixture of words in every topic.

Estimators for topic distributions per text $\boldsymbol{\theta}_m = (\theta_{m,1}, \ldots, \theta_{m,K})^T \in (0,1)^K$ and word distributions per topic $\boldsymbol{\phi}_k = (\phi_{k,1}, \ldots, \phi_{k,V})^T \in (0,1)^V$ can be derived through the Collapsed Gibbs Sampler procedure (Griffiths and Steyvers, 2004) by

$$\hat{\theta}_{m,k} = \frac{n_k^{(m\bullet)} + \alpha}{N^{(m)} + K\alpha}, \quad \hat{\phi}_{k,v} = \frac{n_k^{(\bullet v)} + \eta}{n_k^{(\bullet\bullet)} + V\eta}.$$

2.2 LDAPrototype

The Gibbs sampler in the modeling procedure of LDA is sensitive to the random initialization of topic assignments. To overcome this issue, the selection algorithm LDAPrototype can be used. The method selects the LDA as prototype model of a set of LDAs that maximizes its mean pairwise similarity to all other models (Rieger et al., 2020). Thus, the LDAPrototype method increases the reliability of conclusions drawn from the resulting prototype model. The approach is implemented in the R package ldaPrototype (Rieger, 2020).

3 Methods

We propose the method RollingLDA that uses preceding LDA results as an initialization for subsequent time intervals. The method builds on an existing implementation of LDA (Chang, 2015) and aims to ensure consistent time series based on textual data. The method provides a memory parameter to use a different number of time units of the past as initialization to find a good trade-off of consistency and flexibility of topics. Different values for the memory parameter can be investigated quantifying topic-self-similarities over time. The method is implemented and published as R package rollinglda (Rieger, 2021) and its source code can be retrieved at `https://github.com/JonasRieger/rollinglda`.

3.1 RollingLDA

A pseudocode of the general method RollingLDA can be found in Algorithm 1. The method has the usual parameters of an LDA: the `corpus` to be modeled, the number of topics modeled `K`, the Dirichlet parameters α, η and the number of iterations `iter`. In addition, there are method specific parameters `chunks`, `memory`, and `limit`. Additionally, in line 4 it is recommended to choose a reliable method for the initial LDA, e.g. LDAPrototype described in Sect. 2.2. In line 9, and throughout this paper, we distinguish between the two possibilities that the assignments to previous documents remain fixed or, alternatively, that they are able to change. In the latter case, the assignments to previous documents are changed only for this specific sequential fitting, but not for the final model.

The parameter `chunks` is used to cut the data into intervals. It is a vector of dates that contains in the first entry the date of the first day of the sequential fitting, i.e. the last day of the initial fitting plus one day. The next entries specify the first days of the corresponding sequential chunks, and the last entry specifies the day of the last observed document plus one day. In the analysis, we choose these dates on an equidistant monthly or quarterly basis. The vector `memory` allows flexible choices of the method's memory in the context of sequential fitting. It determines how much knowledge from modeled texts from the previous chunk(s) is used to model the new chunk/subcorpus. The corresponding vector specifies from which date previous documents are (equally weighted) considered for the current chunk. All We also choose this parameter in this paper on an equidistant basis, considering a fixed number of one to four quarters as memory. The method's implementation also allows to set these date vectors explicitly.

The parameter `limit` consists of a combination of rules for determining the sequential vocabulary.

Algorithm 1: Fitting a RollingLDA model.

Input : `corpus, K, α, η, iter, chunks, memory, limit`
Output : RollingLDA model

begin
  determine `subcorpus`: filter `corpus` to documents published before `chunks[1]`;
  determine `vocab`: words that occur more than `limit` times in `subcorpus`;
  fit LDA on `subcorpus` with parameters `K, α, η, iter, vocab`;
  for `i=1` **to** `length(chunks)-1` **do**
    determine `subcorpus`: filter `corpus` to documents published on or after `chunks[i]` and before `chunks[i+1]`;
    update `vocab`: add words that occur more than `limit` times in `subcorpus`;
    determine `init`: *tabulate* assignments of words to topics for fitted documents published on or after `memory[i]` and before `chunks[i]`; *sample* assignments of words to topics for new documents in `subcorpus`;
    fit LDA on `subcorpus` with parameters `K, α, η, iter, vocab` and `init`;
  end
  determine `result`: combine sequential fittings to one object;
  return `result`
end

For the initial LDA as well as for each subcorpus of documents the vocabulary exceeding a given combination of thresholds is determined (see Sect. 5.2). The vocabulary is monotonically increasing, i.e. previously considered words remain included, such that no information is lost, when time evolves.

In Sect. 5, the RollingLDA method is applied to an example dataset.

3.2 Similarity Measures

Self-similarities of topics over time are useful as indicators for the stability of topics. They can also be used as criteria for the individual choice of the memory parameter of the RollingLDA to ensure flexible and reliable topics. Using the notation from Sect. 2.1 the word count vector for topic $k = 1, \dots, K$ is given by

$$\boldsymbol{n}_k = \left(n_k^{(\bullet 1)}, \dots, n_k^{(\bullet V)}\right)^T \in \mathbb{N}_0^V.$$

Extending the notation to account for different temporal aggregations t leads to $\boldsymbol{n}_{k|t}$. We do not consider the similarity of two different topics (different k) in this paper, but always similarities of the same topic (same k) at different times. Since k is constant within our similarity calculations, we simplify the notation for clarity to

$$\boldsymbol{n}_{k|t} = \boldsymbol{n}_t = (n_{t,1}, \dots, n_{t,V})^T,$$
$$\boldsymbol{p}_t = (n_{t,1}, \dots, n_{t,V})^T \Big/ \sum_v n_{t,v}.$$

We consider two different types of similarity measures: one based on word count vectors $\boldsymbol{n}_i, \boldsymbol{n}_j$, one based on word distribution vectors $\boldsymbol{p}_i, \boldsymbol{p}_j$. Then, cosine similarity and a thresholded version of the Jaccard coefficient, respectively, are defined as

$$\cos = \frac{\sum_v n_{i,v} n_{j,v}}{\sqrt{\sum_v n_{i,v}^2}\sqrt{\sum_v n_{j,v}^2}}, \tag{1}$$

$$\text{TJ} = \frac{\sum_v \mathbb{1}_{\{n_{i,v} > c_i \,\wedge\, n_{j,v} > c_j\}}}{\sum_v \mathbb{1}_{\{n_{i,v} > c_i \,\vee\, n_{j,v} > c_j\}}}. \tag{2}$$

The distributional similarity measures based on the Manhattan, χ^2 and Hellinger distance and Jensen Shannon divergence, respectively, are given by

$$\text{MH} = 1 - \frac{1}{2}\sum_v |p_{i,v} - p_{j,v}|, \tag{3}$$

$$\chi^2 = 1 - \frac{1}{2}\sum_v \frac{(p_{i,v} - p_{j,v})^2}{p_{i,v} + p_{j,v}}, \tag{4}$$

$$\text{HL} = 1 - \sqrt{\frac{1}{2}\sum_v \left(\sqrt{p_{i,v}} - \sqrt{p_{j,v}}\right)^2}, \tag{5}$$

$$\text{JS} = 1 - \sum_v p_{i,v} \log \frac{2p_{i,v}}{p_{i,v} + p_{j,v}} - \sum_v p_{j,v} \log \frac{2p_{j,v}}{p_{i,v} + p_{j,v}}. \tag{6}$$

The thresholds c_i, c_j for TJ may be chosen as an absolute, relative or as combination of both lower bounds. In this paper, we use the default value

$c_{\text{rel}} = 0.002$ as proposed by Rieger et al. (2020). For numerical reasons a small value $\epsilon = 10^{-6}$ is added to the word counts $\boldsymbol{n}_t$ before calculating $\boldsymbol{p}_t$ to determine the similarity using χ^2 and JS.

4 Stability and Sensitivity Analysis

For a brief demonstration of which of the presented similarity measures is particularly well suited for the present case of comparing topics at different points in time, we use Zipf's law (Piantadosi, 2014). This states that for an ordered list of V entries, such as words in this example, the relative frequency of the element with rank r can be written as

$$\frac{1/r^s}{\sum_{v=1}^{V}(1/v^s)}.$$

We consider how stable the similarity measures are in the uncertainty scenario and how sensitive they are to detect strong changes in the topics.

4.1 Simulation Setup

In the present case, we choose $s = 1$ for simplicity, we assume the vocabulary size to be $V = 10\,000$ and observe a total number of $7\,500$ word appearances. Then, with respect to Zipf's law, we set the absolute frequencies of the ten most frequent words as 766, 383, 255, 191, 153, 128, 109, 96, 85, 76.

Taking these frequencies as a snapshot of a topic's assignments at one time interval, we modify certain parts of these frequencies to simulate different events or structural breaks in this topic:

a) A new topic like the Covid pandemic is attached to an existing topic,
b) the frequency of a previously prominent subtopic in a topic de-/increases,
c) the frequency of one previously prominent word in a topic de-/increases.

In addition, we compare various idealistic and rather technical modifications to the frequency vector, namely

d) resampling the frequency vector based on the relative frequencies,
e) shuffling the whole frequency vector,

as well as shuffling only the frequencies of the

f) top 10 words,
g) top 50 words,
h) top 100 words,
i) words ranked at position 11 to 20,
j) words ranked at position 21 to 50.

In this setup, we expect scenario e) to result in the lowest similarity for each similarity measure, because it corresponds to comparing two completely different word frequency vectors, i.e. topics. In contrast, scenarios d), i) and j) should lead to minimal to modest differences (at less important ranks) of the frequency vector and therefore should result in the highest similarities, assuming a well suited similarity measure.

4.2 Findings

In Figure 1, in the first row, we set the last (i.e. least mentioned) 1 to 20 words to an increased frequency (up to 750), and study the effect on the self-similarity of the topic. This fits to scenario a). In the second and third row, the frequencies of the top-ranked words are changed. While in the second row, the first x words are considered, in the third row only the x-th single word's frequency changes. Note that these two rows are scaled on a logarithmic axis: a value of -6 is equivalent to setting the word's frequency to zero, while a value of 4 means multiplying it by $\exp(4) \approx 54.6$.

For the addition of new words, the behavior of all measures is comparable. The Jensen-Shannon similarity shows a slightly lower sensitivity. Manhattan and χ^2 similarities show higher similarities for the addition of only one word than cosine and Hellinger, which already show a stronger effect on the similarity by adding a few words. The most striking characteristic in scenario b) is shown by the cosine similarity. In Figure 1, in the second row, it can be seen that the cosine similarity strongly depends on the top words frequencies. Specifically, by setting the ten most frequent words to zero, the cosine similarity decreases very strongly (to about 0.25), while increasing these top ten words frequencies has almost no effect (similarity close to 1).

At the same time, for all other similarity measures, we observe that increasing the top word frequencies leads to a stronger decrease in similarity than eliminating these top words. In general, all similarity measures show a similar trend for the change of single top ranked words. However, the top word has a particularly strong influence using the cosine similarity. This is plausible, since cosine similarity can be interpreted as the angle between the compared frequency vectors and this angle also strongly depends on the top word's frequency under consideration of Zipf's law.

In Figure 2, the similarity measures for the other introduced scenarios are shown comparatively. The scenarios d), i), j), f), g), h) and e) are shown from left to right for each similarity measure as

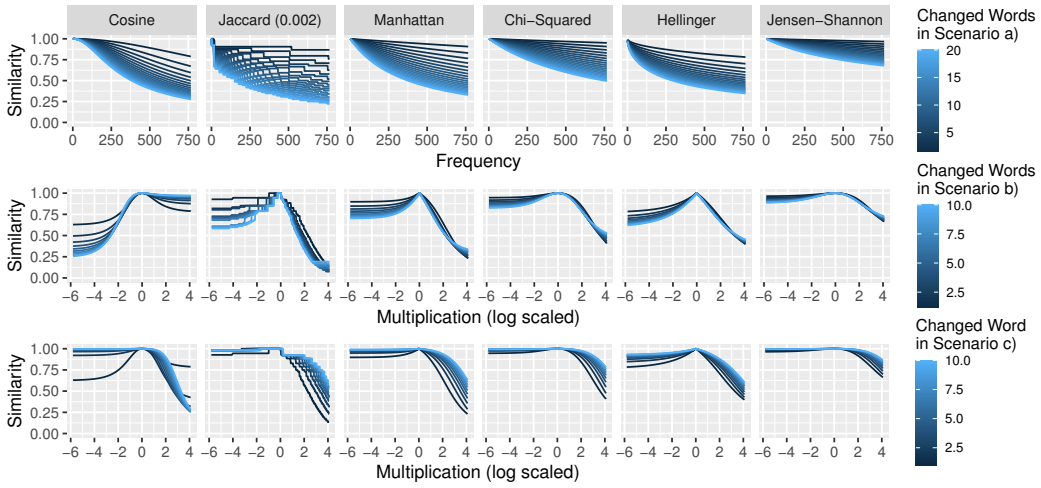


Figure 1: Comparison of similarity measures regarding the effect on topic-self-similarity obtained by modifications of the original word frequency vector with respect to scenario a), b) and c).

this should correspond to the natural decreasing order of the values. Each scenario is based on 500 replications.

Since scenario e) generates strongly different topics, the similarity among them should be as low as possible. This requirement is met by all measures except Jensen-Shannon similarity, but this could made to behave similar as χ^2 or Manhattan by re-scaling. Another desired property is that the uncertainty in word frequencies does not result in dissimilarity. Only cosine similarity satisfies this. For all other measures, statistical uncertainty largely results in greater dissimilarities than modifications from scenarios f), g), i) and j). In real problems, this property can lead to events being masked by variation or, conversely, variation being interpreted as events.

4.3 Use Case and Conclusion

Figure 3 shows the self-similarities of a topic from a RollingLDA model with selected parameters. The topic is about health, so the similarity remains stable in the long term, but has a few shocks in the self-similarity that result from sudden events, such as the Covid outbreak at the beginning of 2020. In Table 1, the five most informative words for selected quarters that realize a quarterly cosine self-similarity less than 0.9 are given. Based on the evolving topwords within the different quarters, events in the corresponding topic can be anticipated, which in particular map the corresponding time series of quarterly cosine self-similarites. The

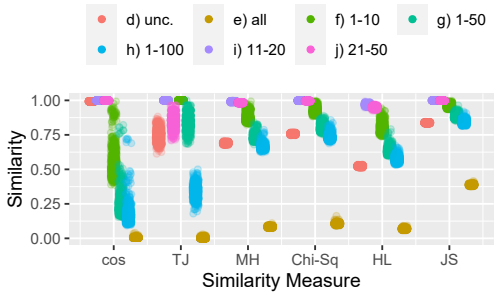


Figure 2: Comparison of similarity measures concerning the effect on self-similarity obtained by applying scenarios d) to j) to the original word frequency vector.

values of the other similarity measures run mostly in parallel, but do not show large differences at key events. In contrast, the Jaccard coefficient seems too sensitive and leads to similarity values that are unstable over time.

In conjunction with the findings from Figures 1 and 2 we recommend to use cosine similarity for the use case of monitoring topic stability or topic-self-similarities, respectively. In addition, in Sect. 5 we mostly stick to quarterly self-similarities as the most appropriate unit.

5 Analysis

In the following, the proposed method RollingLDA is applied to an example dataset. The calculations were performed using R (R Core Team, 2021).

	Overall	1982/Q4	2001/Q4	2002/Q1	2003/Q2	2003/Q3	2014/Q4	2020/Q1
1	dr	dr	anthrax	anthrax	sars	sars	ebola	coronavirus
2	patients	clark	mail	cloning	disease	fasting	duncan	virus
3	disease	tylenol	cipro	aventis	cases	dr	quarantine	outbreak
4	health	clarks	spores	ovarian	respiratory	anemia	sierra	quarantine
5	cancer	capsules	bioterrorism	mammograms	heymann	brain	west	health

Table 1: Time varying topwords of the topic *Health* in the scenario of quarterly modeling with three quarters memory and starting with the rolling approach in 1985 for selected quarters.

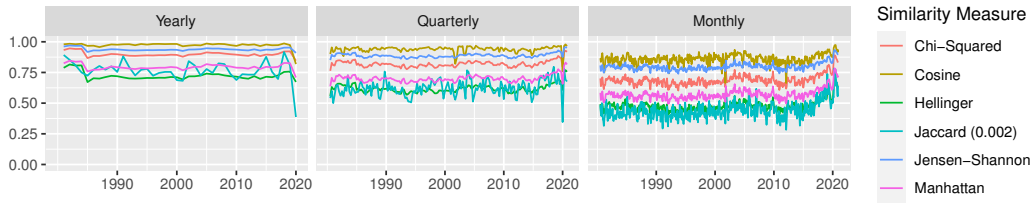


Figure 3: Unit-to-unit self-similarities of the topic *Health* in the scenario of quarterly modeling with three quarters memory and starting with the rolling approach in 1985 for the six different similarity measures.

5.1 Data

The dataset consists of all published articles from The New York Times from June 1, 1980 to December 31, 2020. It was retrieved through the Nexis service (LexisNexis, 2021) and consists of 4 287 928 documents. After applying common natural language processing (NLP) steps such as changing all words to lowercase and stopword removal using the R packages tosca (Koppers et al., 2020) and tm (Feinerer et al., 2008), as well as duplicate removal, 3 767 047 non-empty documents remain in the relevant dataset.

Maier et al. (2020) showed that for datasets of 230 000 documents or more already using at least 10% of the articles results in sufficiently similar topics to the complete dataset. Thus, for a faster calculation, we use a partial dataset for the study. To do this, we draw 15% of all articles stratified by week. This results in a dataset of 566 050 documents with an average of 267 (min: 106, max: 584) documents per week. We also prove the computability on the complete dataset with an exemplary parameter combination.

5.2 Scenarios

Different scenarios are compared to investigate the effects on topic stability and sensitivity. For all cases, we choose as parameters for LDA $K = 80$, $\alpha = \eta = 1/K$ and iterate the Gibbs sampler for 200 iterations. For initial modeling, we use the LDAPrototype method described in Sect. 2.2 with default setting (Rieger, 2020), i.e., in particular,

start	memory	non-changing quarter	non-changing year	changing quarter	changing year
1981	4	7.95	4.75	60.57	23.21
	3	7.78	4.67	48.65	19.98
	2	7.55	4.76	37.56	17.32
	1	7.43	4.58	26.37	14.72
1985	4	7.66	4.43	54.66	21.37
	3	7.37	4.40	44.86	20.87
	2	7.20	4.39	34.64	18.08
	1	7.01	4.30	24.53	15.47
2000	4	5.45	3.22	36.46	16.15
	3	5.35	3.20	29.96	14.29
	2	5.58	3.17	23.28	12.40
	1	5.22	3.21	16.67	10.65

Table 2: Runtime of the RollingLDA models in hours.

the prototype is chosen from $n = 100$ models. In addition, we consider three different time horizons for the initial model: all documents from 1980, 1980–1984, or 1980–1999.

For the parameter `chunks`, we distinguish between quarterly or annual intervals, and for the parameter `memory` between one to four quarters as memory. We choose a combination of relative and absolute threshold as (fixed) `limit` parameter to minimize the disadvantages of both. Words that occur more than five times and cover more than 10ppm of the total word count in a chunk are added, as well as words that simply occur more than 100 times. In addition, we consider the two variants of sequential LDA in line 9 of Algorithm 1, one with fixed, and one with changing previous assignments.

In Table 2 the runtimes of the resulting 48 different models are given. The RollingLDA model

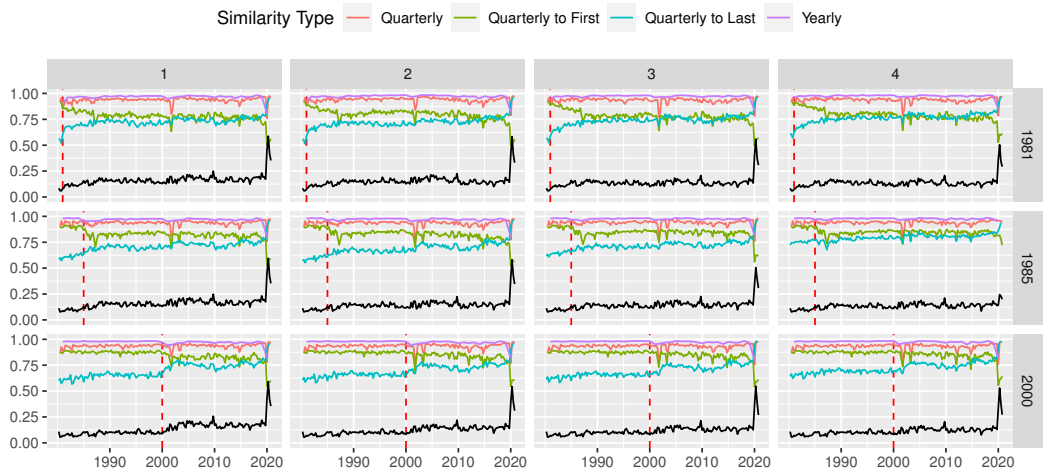


Figure 4: Cosine self-similarities of the topic *Health* for all parameter combinations of the memory and the rolling starting date in the quarterly modeling scenario (topic's scaled share is multiplied by 7 and visualized in *black*).

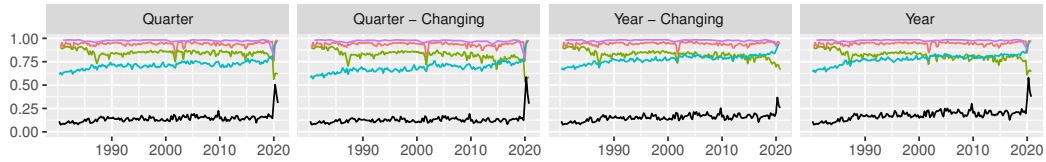


Figure 5: Cosine self-similarities and scaled share of the topic *Health* for *non-changing* and *changing* previous assignments. The scenario of *quarterly* and *yearly* modeling with three quarters memory and starting with the rolling approach in 1985 is considered.

on the complete dataset in the scenario of quarterly modeling with unchanging previous assignments, three quarters memory and starting in 1985 lasts around 48 hours, which meets the assumption of a linear runtime depending on the number of documents. In all analyses, unless explicitly mentioned, the RollingLDA method with non-changing previous assignments is considered.

5.3 Findings

Figure 4 shows the cosine topic-self-similarities for a selected topic *Health* depending on different parameters. A strong topic-self-similarity is noticeable until the start of the sequential modeling. In common applications this is a desired property. In the present case, however, one would like to detect dissimilarities over time. The time series suggest that our method is suitable for this purpose. While the topic seems to remain basically similar, it changes sufficiently from unit to unit and over longer periods of time, which allows the detection of events (cf. Table 1 and Figure 3). The choice of the memory parameter seems to have an intuitive effect, i.e., larger memory tends to lead to stronger anchoring to the past.

As a complement, both the quarterly and annual modeling intervals with non-changing and changing previous assignments are shown in Figure 5 for the special case of three quarters of memory and sequential start in 1985. Here it can also be seen that simultaneous modeling of larger intervals leads to more similar topics over time. In addition, we could not find a substantial difference between changing and non-changing previous assignments (also when looking at other models and topics).

Finally, Figure 6 shows different plausible patterns of topic-self-similarity in the data. There are topics that are very stable overall, but show events (for example *Health*), topics that are very stable overall, show no clear events, but undergo gradual steady change (for example *Technology*), and topics that are taken over by other topics, such as in this case a stopword topic that almost completely disappears. The latter may happen, e.g. when topics

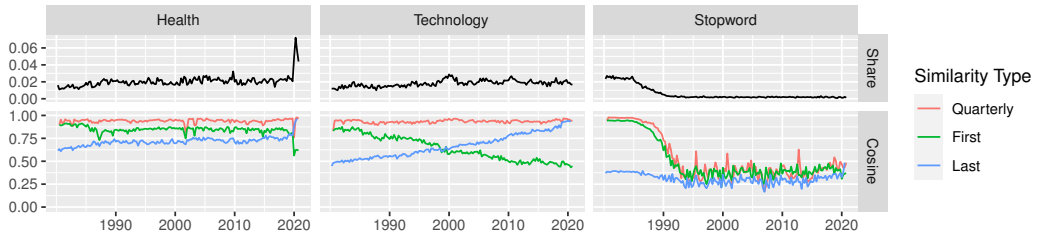


Figure 6: Different patterns of topic's cosine self-similarity for the topics *Health*, *Technology* and a *Stopword* topic in the scenario of quarterly modeling with three quarters memory and starting with the rolling approach in 1985. In addition, the quarterly share of the respective topic is shown in the upper row.

that are sufficiently similar gain in similarity by restricting texts to short(er) intervals, because minor differences (e.g., choice of stopwords in sports articles, choice of stopwords in politics articles) in individual intervals may become so marginal that merging the topics becomes useful in terms of optimizing likelihood in the fitting procedure.

In addition to the mentioned results, we were also able to identify some other patterns in the data, such as seasonal sports topics, which can be found - along with additional analyses - in the associated GitHub repository https://github.com/JonasRieger/emnlp2021.

6 Discussion

We presented a method RollingLDA to model consistent time series from textual data, which is also suitable for monitoring applications due to its efficiency. In particular, it is possible to choose very frequent update intervals and thus to keep the runtime of each update very short.

Apparently, the specific parameterization is not that important, the model seems relatively robust. It is less sensitive with respect to its parameter choice, so that even for more inappropriate parameter choices, the model produces plausible results. Our study has shown, for example, that there is no strong difference between changing previous assignments and fixing previous assignments. However, the latter has a considerable runtime advantage, because the Gibbs sampler does not have to iterate over the previous assignments (the memory) in each time step. For runtime reasons, we therefore recommend the version with non-changing previous assignments.

We also recommend to choose the memory parameter reasonably. It is an important and intuitive parameter, which specifies how much (modeled) past the model takes into account for modeling the next chunk. For example, three quarters of memory in a quarterly modeling scenario means the consideration of one year for each modeling step. When choosing this parameter, one should consider seasonalities, because a topic that only appears in summer, for example, could disappear repeatedly due to a memory that only lasts for one quarter. In case of reappearance it is then not ensured that it receives the same index. Instead, it joins the most similar topic, so that the coherent interpretation of the topic can not be guaranteed.

In addition, the initial LDA should cover a time horizon as short as reasonable, so that a large part of the time series is covered by the rolling approach and can be interpreted accordingly. We also tested sequential prototypes instead of sequential LDAs (cf. line 9 in Algorithm 1). However, it turned out that the set of possible LDAs is very similar such that we observed no further practical gain using the LDAPrototype for each sequential LDA step.

Further research could include weighting the previous documents for the memory or looking at a random sample of those. For the latter case, the consideration of reliable methods for the determination of the update states then again could be interesting. In the long term, one goal is to extend the method to varying numbers of topics per time interval.

Acknowledgements

The present study is part of a project of the Dortmund Center for data-based Media Analysis (DoCMA). In addition, the authors gratefully acknowledge the computing time provided on the Linux HPC cluster at TU Dortmund University (LiDO3), partially funded in the course of the Large-Scale Equipment Initiative by the German Research Foundation (DFG) as project 271512359.

Ethical Considerations

In Sect. 5.1, we explain how we draw a representative sample of the full data for the method comparison. We do this without losing the validity of the results and in order to consider resource efficiency in the context of climate change (Strubell et al., 2019). We also show the efficient feasibility of the method on the full data set as an example.

Reproducibility

All described methods and analyses are provided in the associated GitHub repository `https://github.com/JonasRieger/emnlp2021` together with further graphics for all models. As far as legally possible, the data sets used are also available in this repository. The proposed method is implemented and published as R package, the source code can be retrieved at `https://github.com/JonasRieger/rollinglda`.

References

Nikolaos Aletras and Mark Stevenson. 2014. Measuring the similarity between automatically generated topics. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 22–27, Gothenburg, Sweden. Association for Computational Linguistics.

Hesam Amoualian, Marianne Clausel, Eric Gaussier, and Massih-Reza Amini. 2016. Streaming-LDA: A copula-based approach to modeling topic dependencies in document streams. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 695–704, New York, NY, USA. Association for Computing Machinery.

David M. Blei and John D. Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 113–120, New York, NY, USA. Association for Computing Machinery.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Jonathan Chang. 2015. *lda: Collapsed Gibbs Sampling Methods for Topic Models*. R package version 1.4.2.

Jonathan Chang, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. 2009. Reading Tea Leaves: How Humans Interpret Topic Models. In *NIPS: Advances in Neural Information Processing Systems*, pages 288–296. Curran Associates Inc.

Ingo Feinerer, Kurt Hornik, and David Meyer. 2008. Text Mining Infrastructure in R. *Journal of Statistical Software*, 25(5):1–54.

Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235.

Bettina Grün and Kurt Hornik. 2011. topicmodels: An R Package for Fitting Topic Models. *Journal of Statistical Software*, 40(13):1–30.

Matthew Hoffman, Francis Bach, and David Blei. 2010. Online learning for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.

Paul Jaccard. 1912. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50.

Nathan Keane, Connie Yee, and Liang Zhou. 2015. Using topic modeling and similarity thresholds to detect events. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 34–42, Denver, Colorado. Association for Computational Linguistics.

Dongwoo Kim and Alice Oh. 2011. Topic Chains for Understanding a News Corpus. In *CICLing: Computational Linguistics and Intelligent Text Processing*, volume 6609 of *LNCS*, pages 163–176. Springer.

Lars Koppers, Jonas Rieger, Karin Boczek, and Gerret von Nordheim. 2020. *tosca: Tools for Statistical Content Analysis*. R package version 0.2-0.

Solomon Kullback and Richard A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

LexisNexis. 2021. Nexis: LexisNexis Academic & Library Solutions.

Jianhua Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

Daniel Maier, Andreas Niekler, Gregor Wiedemann, and Daniela Stoltenberg. 2020. How document sampling and vocabulary pruning affect the results of topic models. *Computational Communication Research*, 2(2):139–152.

Daniel Maier, A. Waldherr, P. Miltner, G. Wiedemann, A. Niekler, A. Keinert, B. Pfetsch, G. Heyer, U. Reber, T. Häussler, H. Schmid-Petri, and S. Adam. 2018. Applying LDA Topic Modeling in Communication Research: Toward a Valid and Reliable Methodology. *Communication Methods and Measures*, 12(2-3):93–118.

David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Proceedings of the 2011 Conference on Empirical*

Methods in Natural Language Processing, pages 262–272, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Viet-An Nguyen, Jordan Boyd-Graber, and Philip Resnik. 2014. Sometimes average is best: The importance of averaging for prediction using MCMC inference in topic modeling. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1752–1757, Doha, Qatar. Association for Computational Linguistics.

Steven T. Piantadosi. 2014. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin & Review*, 21:1112–1130.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Jonas Rieger. 2020. ldaPrototype: A method in R to get a Prototype of multiple Latent Dirichlet Allocations. *Journal of Open Source Software*, 5(51):2181.

Jonas Rieger. 2021. *rollinglda: Construct Consistent Time Series from Textual Data*. R package version 0.1.0.

Jonas Rieger, Jörg Rahnenführer, and Carsten Jentsch. 2020. Improving Latent Dirichlet Allocation: On Reliability of the Novel Method LDAPrototype. In *NLDB: Natural Language Processing and Information Systems*, volume 12089 of *LNCS*, pages 118–125. Springer.

Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. 2012. Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 952–961, Jeju Island, Korea. Association for Computational Linguistics.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Chong Wang, David Blei, and David Heckerman. 2008. Continuous time dynamic topic models. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, UAI'08, pages 579–586, Arlington, Virginia, USA. AUAI Press.

Xuerui Wang and Andrew McCallum. 2006. Topics over time: A non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 424–433, New York, NY, USA. Association for Computing Machinery.

Yu Wang, Eugene Agichtein, and Michele Benzi. 2012. TM-LDA: Efficient online modeling of latent topic transitions in social media. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 123–131, New York, NY, USA. Association for Computing Machinery.

Yunli Wang and Cyril Goutte. 2018. Real-time change point detection using on-line topic models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2505–2515, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Guixian Xu, Yueting Meng, Zhan Chen, Xiaoyu Qiu, Changzhi Wang, and Haishen Yao. 2019. Research on topic detection and tracking for online news texts. *IEEE Access*, 7:58407–58418.

Ke Zhai and Jordan Boyd-Graber. 2013. Online latent Dirichlet allocation with infinite vocabulary. In *Proceedings of the 30th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 561–569, Atlanta, Georgia, USA. PMLR.

LDAPrototype: A Model Selection Algorithm to Improve Reliability of Latent Dirichlet Allocation

Jonas Rieger[1*], Carsten Jentsch[1†] and Jörg Rahnenführer[1†]

[1*]Department of Statistics, TU Dortmund University, 44221 Dortmund, Germany.

*Corresponding author(s). E-mail(s):
rieger@statistik.tu-dortmund.de;
Contributing authors: jentsch@statistik.tu-dortmund.de;
rahnenfuehrer@statistik.tu-dortmund.de;
†These authors contributed equally to this work.

Abstract

For understanding large text corpora, a widely used method is Latent Dirichlet Allocation (LDA). The topic assignments from LDA usually rely on a (random) initialization such that the outcome is also to some extent random. In particular, replicated runs on the same text data lead to different results such that the LDA is not fully reproducible. This *instability* of the LDA approach is often neglected in practice, where text data analysis is commonly based only on a single LDA run. We propose a new method called LDAPrototype for selecting the most representative run from a set of replicated LDA runs on the same dataset. We improve the *reliability* of conclusions drawn from LDA results, since replications of LDAPrototype are more similar to each other than replications of single LDA runs, or even as models selected by perplexity or NPMI. For this purpose, we propose a reliability score based on a new tailored model similarity measure S-CLOP. For the quantification of topic similarities, we compare a thresholded version of the Jaccard coefficient to cosine similarity, Jensen-Shannon divergence and Rank Biased Overlap. Our results of an application to six real datasets consisting of newspaper articles or tweets show that the reproducibility of LDA results using LDAPrototype increases, and so does the reliability of empirical findings based on topic modeling. Overall, the new approach is justified in view

of its application, comprehensibility, easy implementation and computational feasibility. The algorithm's concept is generalizable to other topic modeling procedures with topics characterized by word distributions.

Keywords: topic model, stability, similarity, medoid, clustering, replications

1 Introduction

Understanding unstructured data is generally a big challenge due to their complicated and typically non-standard forms. In particular, when dealing with text data, this is also the case due to the increasingly large volumes collected steadily these days. Text data is clearly one of the most frequent data types in the world today, and text mining tools have become very popular to analyze such data with the goal to address research questions from various fields of research.

Text data are often analyzed using so called probabilistic topic models [1] and the Latent Dirichlet Allocation [2] in particular. In the context of probabilistic topic models, the Latent Dirichlet Allocation (LDA) can be described as a successor to the Probabilistic Latent Semantic Indexing [3] and a predecessor to the Correlated Topic Models [4]. The LDA model has been extended by numerous features in the past years. Many of the implementations aim at the integration of meta variables such as author names in the Author-Topic Model [5]. Other models have attempted to integrate the temporal component of the modeled texts, e.g. the Continuous Time Dynamic Topic Models [6]. In addition to the further development of the LDA, the Structural Topic Model [7] has proven to be another reliable model for the analysis of text corpora. Nevertheless, LDA still enjoys the highest popularity due to its simple implementation, flexible assumptions, and competitive results compared to other more refined, but also more complex methods.

In this paper, we propose the method LDAPrototype to improve the reliability of LDA results. In several use cases we show that single models are prone to misinterpretation because of potentially large differences in the results of LDA runs executed for the same text data. To do this, we define a similarity measure for LDA models, which we call S-CLOP. This measure can be used to determine pairwise similarities of LDA models. To overcome the mentioned issue of reproducibility, which in turn leads to lower reliability, the method LDAPrototype selects the most representative run out of a set of LDA runs according to the novel S-CLOP criterion. This criterion is not based on likelihood-based measures as it is often the case. Instead, our goal is to determine the medoid of the models to increase the reliability of conclusions drawn from LDA results. In this context, we understand reliability as a measure to quantify the reproducibility of the results. That is, a highly reliable method should produce results that are as reproducible as possible. We refer to the medoid as the model that agrees most on average with all other models. We

call the resulting model the prototype, which is most similar to all other runs taken from the same modeling procedure obtained from he same text data.

A few approaches exist to make LDA results more reliable, but all of them have weaknesses, which are discussed in detail in Section 2.2. Often the modeling procedure itself is influenced in a way such that LDA loses its flexibility. Other methods do not search in the whole space of possible models, which may lead to non-optimal results. To address these weaknesses, our approach does explicitly not affect the modeling procedure itself and is exclusively based on replicated runs of the same topic model.

We do not use likelihood-based measures, because there is already good comparative work in this field, e.g. by [8], and [9], who showed that these measures do not correlate well with the human perception of consistent topics. Nevertheless, we evaluate the ability of our method to find reliable models in comparison to the relatively popular measure of perplexity and the well-known coherence measure NPMI. However, the quality of a topic model, which is mainly argued from a high interpretability of the results, is not within the scope of this paper. Subsequent studies with human judgments are needed for this. Instead, the main goal is to address the issue of reliability caused by the instability inherent to the LDA through a model selection criterion to increase the reliability of the results. For this purpose, we define reliability as a measure of the degree of reproducibility. We call a model highly reliable if it produces very similar results under the same model parameter settings, such that the resulting models are only differing due to the random initializations. In Section 3, we formally define a measure of reliability, for which we then show that our new method LDAPrototype achieves higher scores than common selection methods based on perplexity or NMPI.

High reliability can be considered as the basis and crucial prerequisite to also obtain good results in terms of quality. Thus, we do not follow the classical information retrieval approach, but focus first on increasing reliability of results in order to finally obtain topics that can be interpreted particularly well, which then would correspond to model's quality. We intentionally do not speak of an improvement in stability here, because the LDA method, which is intrinsically depending on a random initialization, is not changed, and thus is just as (in)stable as before. Our proposed selection algorithm based on replicated runs on the same text data to obtain the most representative LDA run, on the contrary, provides an improvement of the reliability, which is measured by the degree of reproducibility quantified by our reliability score.

2 Related Work

Text data are usually organized in large corpora, where each corpus consists of a collection of texts, often also denoted as documents or articles. Each text can be considered as a sequence of tokens of words of the same length as the given text. In common notation token means an individual word at a specific place in the text and the set of words is used synonymously with vocabulary.

We refer to these terms in the following to introduce the methodology of LDA as basis for our novel method LDAPrototype.

2.1 Latent Dirichlet Allocation

The method we propose is based on the LDA [2] estimated by a collapsed Gibbs sampler [8]. The LDA assumes distributions of latent topics for each text. If K denotes the total number of modeled topics, the set of topics is given by $\boldsymbol{T} = \{T_1, \dots, T_K\}$. We define $W_n^{(m)}$ as a single token at position n in text m. The set of possible tokens is given by the vocabulary $\boldsymbol{W} = \{W_1, \dots, W_V\}$ with $V = |\boldsymbol{W}|$, the vocabulary size. Then, let

$$\boldsymbol{D}^{(m)} = \left(W_1^{(m)}, \dots, W_{N^{(m)}}^{(m)}\right),\ m = 1, \dots, M, \quad W_n^{(m)} \in \boldsymbol{W},\ n = 1, \dots, N^{(m)}$$

be text (or document) m of a corpus consisting of M texts, each text of length $N^{(m)}$. Topics are referred to as $T_n^{(m)}$ for the topic assignment of token $W_n^{(m)}$. Then, analogously the topic assignments of every text m are given by

$$\boldsymbol{T}^{(m)} = \left(T_1^{(m)}, \dots, T_{N^{(m)}}^{(m)}\right),\ m = 1, \dots, M, \quad T_n^{(m)} \in \boldsymbol{T},\ n = 1, \dots, N^{(m)}.$$

When $n_k^{(mv)}, k = 1, \dots, K, v = 1, \dots, V$ describes the number of assignments of word v in text m to topic k, we can define the cumulative count of word v in topic k over all documents by $n_k^{(\bullet v)}$ and, analogously, the cumulative count of topic k over all words in document m by $n_k^{(m\bullet)}$, while $n_k^{(\bullet\bullet)}$ indicates the total count of assignments to topic k. Then, let

$$\boldsymbol{w}_k = \left(n_k^{(\bullet 1)}, \dots, n_k^{(\bullet V)}\right)^T \in \mathbb{N}_0^V,\ k = 1, \dots, K$$

denote the vector of word counts for topic k.

Using these definitions, the underlying probability model [8] can be written as

$$\begin{aligned}
W_n^{(m)} \mid T_n^{(m)}, \boldsymbol{\phi}_k &\sim \text{Discrete}(\boldsymbol{\phi}_k),\\
\boldsymbol{\phi}_k &\sim \text{Dirichlet}(\eta),\\
T_n^{(m)} \mid \boldsymbol{\theta}_m &\sim \text{Discrete}(\boldsymbol{\theta}_m),\\
\boldsymbol{\theta}_m &\sim \text{Dirichlet}(\alpha).
\end{aligned}$$

For a given parameter set $\{K, \alpha, \eta\}$, LDA assigns one of the K topics to each token. Here K denotes the number of topics and α, η are parameters of a Dirichlet distribution defining the type of mixture of topics in every text and the type of mixture of words in every topic. Higher values for α lead to a more heterogeneous mixture of topics whereas lower values are more likely to produce less but more dominant topics per text. Analogously, η controls the

mixture of words in topics. Although the LDA permits α and η to be vector valued [2], they are usually chosen symmetric because typically the user has no a-priori information about the topic distributions $\boldsymbol{\theta}$ and word distributions $\boldsymbol{\phi}$.

Topic distributions per text $\boldsymbol{\theta}_m = (\theta_{m,1}, \ldots, \theta_{m,K})^T \in (0,1)^K$ and word distributions per topic $\boldsymbol{\phi}_k = (\phi_{k,1}, \ldots, \phi_{k,V})^T \in (0,1)^V$ can be estimated through the collapsed Gibbs sampler procedure [8] by

$$\hat{\theta}_{m,k} = \frac{n_k^{(m\bullet)} + \alpha}{N^{(m)} + K\alpha} \qquad \text{and} \qquad \hat{\phi}_{k,v} = \frac{n_k^{(\bullet v)} + \eta}{n_k^{(\bullet\bullet)} + V\eta}.$$

2.2 Methods and modifications of LDA to address the random initialization instability

Inferring LDA using Gibbs sampling is sensitive to the initial assignments, that are often chosen as random, and the reassignment is based on the conditional distributions, which leads to different results in multiple LDA runs for fixed parameters. This instability of LDA leads to a lack of reliability of the modeling results. This fact is rarely discussed in applications [10], although several approaches have been proposed to encounter this problem as discussed in the following.

Parameter tuning

[10] propose a new algorithm LDADE (LDA **D**ifferential **E**volution) which automatically tunes the parameters of LDA in order to optimize topic similarity in replications using a differential evolution algorithm. This results in a set of input parameters K, α and η which perform best on the given data with respect to reliability. This procedure does not increase the reliability for a given parameter set, but tries to find the parameter set that produces the most reliable results. So this method aims for parameter optimization. However, it is likely that the tuning algorithm is biased to select parameters that result in systematically better reliability values independent of the underlying dataset, e.g. for low α and η parameters. In many applications, it is of interest to choose the parameters of the LDA reasonably based on external knowledge. Accordingly, our method focuses on the optimization of the reliability for fixed parameter sets instead of parameter optimization as performed by LDADE.

Selection and averaging algorithms

Another option is to apply a selection criterion to a set of models [11]. The selection can be done by optimizing perplexity [2], a performance measure for probabilistic models to estimate how well new data fit into the model [5]. Alternatively, [12] proposed to average stages of the Gibbs sampling procedure. They present different variations to average iteration steps and show that their approach leads to an increase of perplexity. Averaging LDA models comes with the drawback that one only receives averaged topic proportions, but no specific topic assignment per token. In addition, it was shown that likelihood-based

measures like perplexity are negatively correlated with human judgments on topic quality [9]. Instead, optimizing the semantic coherence of topics should be the aim for a selection criterion. [9] provide a validation technique called Word/Topic Intrusion [implemented in 13] which depends on a human coding process. Automated measures to select the best LDA regarding coherence can be transferred from the topic coherence measure [14, 15], but there is no stable and validated aggregation technique of this type of topic quality measure for the results of LDA runs. Instead, [16] introduce some other measures for quantifying topic quality based on coherence measures, such as the normalized pointwise mutual information (NPMI), for which [17] show, that it outperforms other automatically calculated quality measures regarding correlation to human scores for topic quality. For this reason in Section 6.2, we compare the still most popular selection measure perplexity and the coherence measure NPMI in terms of improving the reliability of the selected LDA models.

Manual approaches and reasonable initialization

[18] aim for increasing both, reliability and interpretability of the final model simultaneously. Therefore, they maximize topic similarity as well as topic coherence, but discover that standard metrics in general do not perform well in increasing interpretability. Instead, manual approaches as the mentioned intruder validation technique proposed by [9] are essential. [18] propose to increase reliability of LDA by initializing topic assignments of the tokens reasonably, e.g. using co-occurrences of words [19]. This initialization technique has the drawback that the model is restricted to a subset of possible results.

Model modifications

There is also a modification of the implementation of LDA that aims to reduce instability. GLDA (**G**ranulated LDA) was proposed by [20] and is based on a modified Gibbs Sampler. The idea of the algorithm is that tokens that are closer to each other are more likely to be assigned to the same topic. The authors show that their algorithm performs comparably well with standard LDA regarding interpretability. Moreover, it leads to more stable results. However, their study is based on only three LDA runs and the implementation is not publicly available. Thus, a validation of this method on other datasets or with larger numbers of replications is pending.

2.3 Contributions

In this work, we propose the novel selection algorithm LDAPrototype based on the tailored similarity measure S-CLOP for LDA models. Thus, our contribution is two-fold. The S-CLOP measure is able to assess the stability of LDA with clustering techniques applied to replicated LDA runs. High stability corresponds to high reliability of findings based on stable models in the sense of improving reproducibility. We introduce a new automated method of clustering topics, more precisely a pruning algorithm for results of hierarchical clustering, based on the optimality criterion that for clustered results of

replicated LDA runs, in the ideal case each cluster should contain exactly one topic of every replication of the modeling procedure. This results in our novel tailored similarity measure S-CLOP (**S**imilarity of multiple sets by **C**lustering with **LO**cal **P**runing) for LDA runs. We demonstrate the potential of this measure to improve reliability by applying it to example corpora. Based on the newly proposed similarity measure S-CLOP, we propose a combination of a repetition strategy and selection criterion to increase the reliability of findings from LDA models leading to the LDAPrototype algorithm. We show, that it outperforms perplexity and NPMI regarding a reliability measure that is based on LDA similarities.

3 Methods

We introduce the new method LDAPrototype that selects the medoid of a number of LDA runs. The selection is achieved by choosing the model that maximizes the mean pairwise S-CLOP value to all other LDA runs. For assessing similarities of LDA models using our novel S-CLOP measure, also an adequate similarity measure for topics is required. We define a more robust version of the Jaccard coefficient in the sense that not all words are considered as relevant for each topic. In Section 6, the selection algorithm LDAPrototype is applied to six example corpora to assess the increase in reliability of findings from LDA models. In Section 6.3, we also present other implemented similarity measures, which are compared to our thresholded Jaccard coefficient regarding reliability gain and computation time.

The introduced methods have been implemented as R package [21] on CRAN and are available at https://github.com/JonasRieger/ldaPrototype as continuously developing GitHub repository.

3.1 LDAPrototype: a new selection algorithm for LDA models

We propose the novel selection algorithm LDAPrototype to improve the reliability of LDA results. For this, we define the reliability score rs for a set of L LDAs based on their mean similarities $\bar{s_1}, \dots, \bar{s_L}$ as

$$rs\left(\bar{s_1}, \dots, \bar{s_L}\right) := \frac{1}{L} \sum_{l=1}^{L} \bar{s_l}, \quad \bar{s_l} \in [0, 1], \tag{1}$$

where the mean similarities $\bar{s_l}, l = 1, \dots, L$ have to be determined by a model similarity measure yet to be defined. Our selection algorithm aims for maximizing this reliability score rs, which measures how reproducible LDA results are in a given setting with fixed parameters. Our algorithm is motivated by an increase in reliability (cf. Section 1) and selects from a set of LDA models the model that is most similar on average to all other runs. The approach is

Algorithm 1 Selecting the medoid of a number of LDA runs

Input: A set of R LDA models

Output: The LDA with maximal mean pairwise similarity to all other LDA runs

1: **for** $i = 1$ **to** $R - 1$ **do**
2: **for** $j = i + 1$ **to** R **do**
3: Calculate pairwise similarity of LDAs lda_i and lda_j using a similarity measure sim: $s_{ij} = \mathsf{sim}(\mathsf{lda}_i, \mathsf{lda}_j)$
4: **end for**
5: **end for**
6: Determine $\bar{s_i} = \frac{1}{R-1} \sum_{j \neq i} s_{ij},\ i = 1, \ldots, R$
7: Determine $\mathsf{proto} = \mathsf{lda}_{\text{opt}}$ with $\text{opt} = \underset{i \in \{1,\ldots,R\}}{\arg\max}\ \bar{s_i}$
8: **return** proto

similar to the choice of the median in the one-dimensional space. In the multidimensional space, this choice is called medoid and differs from a centroid in the sense that it is not obtained by model averaging, but by model selection. There are methods of model averaging for LDA (cf. Section 2.2), but these have the disadvantage that properties of a single run, such as the assignments of individual tokens to topics, are lost. The proposed selection algorithm preserves this information because it does not influence the modeling itself. The LDAPrototype procedure selects one single model from the set of candidate models.

Figure 1 shows schematically the determination of the prototype, and the corresponding procedure is presented in Algorithm 1. The main idea of the method is to calculate all pairwise similarities of the R LDAs and determine the model that maximizes this similarity. Besides the candidate set of LDAs $\{\text{LDA}_1, \ldots, \text{LDA}_R\}$, a similarity measure for LDA models is needed to obtain the symmetric model similarity scores $s_{ij} = s_{ji}$ in Figure 1 and Algorithm 1. For this, in Section 3.3, we propose a novel tailored model similarity measure called S-CLOP, which in turn requires the choice of a topic similarity measure. In the following Section 3.2, we propose a default measure for computing these topic similarities. Through these definitions, we are then able to determine the prototype with respect to Figure 1. In Appendix A, we discuss other popular measures for computing topic similarities, which we also compare to our proposed Jaccard coefficient from Section 3.2 in Section 6.3.

3.2 Thresholded version of the Jaccard coefficient: a similarity measure for topics

A similarity between two topics can be calculated based on the corresponding vectors of counts or set of words. We build on the well established Jaccard coefficient [22] and introduce a more robust thresholded version. Its general

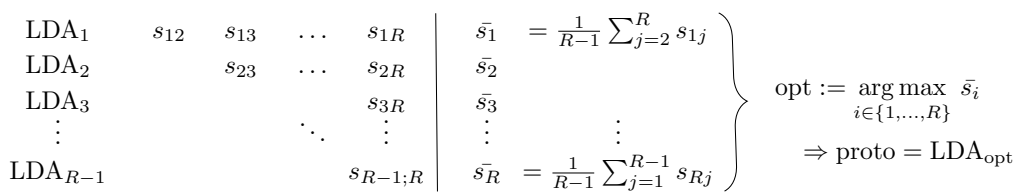


Fig. 1 Schematic representation of the determination of a prototype based on a set of LDA models.

form is given by

$$\text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{2}$$

where A, B are sets of words.

Suppose we have a text corpus and we estimate a topic model with LDA, with parameters α, η and K topics. This is done R times independently leading to a set of $N = RK$ topics in total. Then, referring to Section 2.1,

$$\boldsymbol{w}^{(r)} = \left(\boldsymbol{w}_1^{(r)}, \ldots, \boldsymbol{w}_K^{(r)}\right) \in \mathbb{N}_0^{V \times K}, \; r = 1, \ldots, R$$

denotes the matrix of word counts per topic in the r-th replication. Then, for a given lower bound $\boldsymbol{c} = (c_1, \ldots, c_N)$ and for two topics (i, j) represented by their word count vectors

$$\boldsymbol{w}_i, \boldsymbol{w}_j \in \left\{\boldsymbol{w}_1^{(1)}, \ldots, \boldsymbol{w}_K^{(1)}, \boldsymbol{w}_1^{(2)}, \ldots, \boldsymbol{w}_K^{(2)}, \ldots, \boldsymbol{w}_1^{(R)}, \ldots, \boldsymbol{w}_K^{(R)}\right\}$$

our thresholded version of the Jaccard coefficient is calculated by

$$\text{tJacc}(\boldsymbol{w}_i, \boldsymbol{w}_j) = \frac{\sum_{v=1}^{V} \mathbb{1}_{\left\{n_i^{(\bullet v)} > c_i \;\wedge\; n_j^{(\bullet v)} > c_j\right\}}}{\sum_{v=1}^{V} \mathbb{1}_{\left\{n_i^{(\bullet v)} > c_i \;\vee\; n_j^{(\bullet v)} > c_j\right\}}}. \tag{3}$$

Reasonable choices for the threshold vector $\boldsymbol{c} = (c_1, \ldots, c_N)^T \in \mathbb{N}^N$ are an equal absolute lower bound c_{abs} for all words, or a topic-specific relative lower bound c_{rel} (see also Table 1). A combination of both can be defined by

$$c_l = \max\{c_{\text{abs}}, c_{\text{rel}} n_l^{\bullet\bullet}\},$$

where $l = 1, \ldots K, \ldots, 2K, \ldots RK = N$ and $c_{\text{abs}} \in \mathbb{N}_0, c_{\text{rel}} \in [0, 1]$. To ensure that always enough words per topic are taken, the similarity measure is additionally implemented with a parameter that controls that at least a fixed

Table 1 Toy example: Assignment counts of two topics and calculation of thresholded version of the Jaccard (tJacc) coefficient for $c_{\text{rel}} = 0.002$

	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\wedge$	$\vee$
trump	1 668	2 860	1	1
trumps	446	854	1	1
president	91	876	1	1
donald	259	693	1	1
news	695	0	0	1
said	500	0	0	1
election	8	474	0	1
will	0	462	0	1
women	397	53	1	1
debate	394	11	0	1
sarcastic	1	4	0	0
Σ	4 459	6 287	5	10

vocabulary size $V = 11$,
relative limit $c_{\text{rel}} = 1/500$

$$\Rightarrow \boldsymbol{c} = (n_1^{\bullet\bullet}, n_2^{\bullet\bullet})^T/500 = (4\,459, 6\,287)^T/500 = (8.92, 12.57)^T.$$

$$\text{tJacc}(\boldsymbol{w}_1, \boldsymbol{w}_2) = \frac{5}{10}.$$

user-defined number of the most frequent words assigned to the topic are considered.

The interpretation of this tJacc coefficient is the following. It is defined as the ratio of the numbers of the intersection and the union of the words of two topics, but a word is only considered if the number of its occurrences in the texts exceeds the topic-specific threshold. In other words, we first restrict ourselves to the most relevant words per topic with respect to the number of assignments, to get rid of heavy tailed word lists. Then the resulting subsets of words are used to measure similarity of topics using the standard Jaccard coefficient. To be clear: even for a choice of $c_{\text{rel}} = 0$, the similarities of the topics would not (necessarily) all equal 1. That is because for the calculation not the estimators $\hat{\phi}_{k,v}$, but the numbers of actual assignments $n_i^{(\bullet v)}$ are used.

We demonstrate how the measure is calculated with a small toy example. In Table 1, for eleven selected words the counts of assignments over all articles for the two topics $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are given. We use the relative lower bound with $c_{\text{rel}} = 1/500$. In the analysis presented in Section 6, we mainly also use $c_{\text{rel}} = 1/500$, which leads to around 100 important words per topic in the setting of the *usatoday* dataset. The last two columns indicate whether the corresponding word belongs to the thresholded intersection or union, respectively. For example, the word *election* does not belong to the intersection because its count is below the topic-specific (relative) threshold of at least nine assignments to the topic belonging to $\boldsymbol{w}_1$. The ratio of the number of entries in the third and the fourth column results in the similarity $\text{tJacc}(\boldsymbol{w}_1, \boldsymbol{w}_2) = \frac{5}{10} = 0.5$ of the two given topics.

3.3 S-CLOP: a new similarity measure for LDA models

We introduce the new similarity measure S-CLOP for comparing different LDA runs consisting of topics that are represented by word count vectors. The pairwise distances are calculated with the tJacc coefficient in Equation (3).

However, the measure can be applied using any appropriate distance respectively similarity measure, see Appendix A. We use S-CLOP as pairwise model similarity measure in the LDAPrototype procedure, see also Figure 1 and Algorithm 1.

The general idea of the measure S-CLOP is the following. First, join all considered LDA runs to one overall set, then cluster the topics with subsequent local pruning, and then check how many members of the original different LDA runs are contained in the resulting clusters. Then the deviation from the perfect situation of one representative from each LDA run is quantified. In our selection algorithm always two LDA models are compared. Two models are very similar, if always one topic of the first model is clustered together with one topic from the other model. High S-CLOP similarity values suggest that many topics can be identified that have a representative in each LDA run.

For the initial clustering step, we use hierarchical clustering with complete linkage [23, pp. 520–525]. According to [24] average linkage or Ward's method obtain superior results for the case of text clustering. We also tried single linkage, average linkage and Ward's method in the present case, and in the package the user is also offered, in principle, the possibility of calculation using these link methods. In fact, for the selection algorithm presented here, there is little difference in which of the linkage methods is used due to the characteristic of combining pairwise comparison with local-optimal pruning. We prefer complete linkage over single or average linkage because it uses the maximum distance between topics to identify clusters. This is consistent with our aim of identifying highly homogeneous groups. Of course, topic similarities must first be transformed to distances to apply hierarchical clustering.

Measuring disparity of LDA runs

Consider a cluster (respectively a group) g of topics, after clustering R LDA runs in one joint cluster analysis, using all $R \cdot K$ topics from all runs. The goal is to quantify the deviation from the desired situation that each run is represented exactly once in g. The vector $\boldsymbol{t}^{(g)} = (t_1^{(g)}, \ldots, t_R^{(g)})^T \in \mathbb{N}_0^R$ contains the number of topics that belong to the different LDA runs. Then, we define the disparity measure

$$U(g) := \frac{1}{R} \sum_{r=1}^{R} |t_r^{(g)} - 1| \cdot \sum_{r=1}^{R} t_r^{(g)}. \tag{4}$$

The first factor $|t_r^{(g)} - 1|$ measures the deviation from the best case of exactly one topic per run in g. The second factor determines the number of members in the cluster and is required to penalize large clusters. Without this adjustment, the algorithm presented below for minimizing the sum of disparities would prefer one large cluster over a number of small clusters. In particular, without the second term, joining two perfect clusters as well as splitting one perfect cluster in two clusters would result in the same value for the mean disparity ($R/R = 1$), and we prefer the second situation, where two different topics

Algorithm 2 Determining the minimal sum of disparities $U^*(g)$ of a cluster g

Input: A node of a dendrogram
Output: The minimal possible sum of disparities for this node

```
if is.leaf(node) then
    return (R − 1)/R
else
    return min{U(node), Recall(node.left) + Recall(node.right)}
end if
```

from one run are not clustered together. The disparity of one overall cluster g containing all topics, e.g. defined by the root of a dendrogram, is given by $U(g) = (K-1) \cdot N$.

Finding the best cluster result by minimizing average disparity

The goal is to minimize the sum of disparities $U(g)$ over all groups $g \in G$ of a cluster result. Hierarchical clustering of all N objects (topics) provides cluster results with $1, \ldots, N$ clusters. A common approach is to globally cut the dendrogram according to a target value. Here, we propose to prune the resulting dendrogram locally to obtain the final clusters. The pruning algorithm requires as input a hierarchical clustering result and minimizes the sum of disparities, with respect to the dendrogram structure, i.e.

$$U_\Sigma(G) := \sum_{g \in G} U(g) \to \min, \tag{5}$$

where G is a set of clusters (of topics), and the set of all topics is a disjoint union of the members of the single clusters $g \in G$.

Denote by G^* the optimal set of clusters resulting from splits identified from the dendrogram, and by $U^* := U_\Sigma(G^*)$ the corresponding minimal sum of disparities. The root of the dendrogram contains as a disjoint union the members of the two nodes obtained by the first split. Likewise, iteratively, each node contains as a disjoint union the members of the two nodes on a clustering level one step below this specific node, as denoted in Algorithm 2 by node.left and node.right. The optimal sum U^* can be calculated recursively with Algorithm 2.

For a node in the dendrogram, we denote by $U(\mathsf{node})$ the disparity of the corresponding cluster and by $U^*(\mathsf{node})$ the minimal sum of disparities of the dendrogram induced by (or below) this node. Algorithm 3 can now be used to find the best set of clusters. A cluster is added to the list of final clusters, if its disparity is lower than every sum of disparities obtained when further splitting this node.

Algorithm 3 Finding the optimal set of clusters G^*

Input: A dendrogram with a root

Output: A list correponding to the optimal set of clusters G^*, obtained by local pruning of the dendrogram node = root

```
if U(node) == U*(node) then
    Add all objects belonging to the cluster corresponding to node as one cluster to list
else
    Recall(node.left)
    Recall(node.right)
end if
return list
```

Measuring similarity with aggregated disparities

Finally, we can calculate the similarity of a set of LDA runs using the optimized set of clusters. We normalize the sum of disparities of the optimal clustering, such that its values lie in the interval $[0, 1]$, where 0 corresponds to the worst case and 1 to the best case. The worst case is a pruning state with R clusters, each consisting of all topics from one LDA run. Then the pruning of Algorithm 3 would lead to a set $\tilde{G}$ of N single topic clusters, resulting in the highest possible value for the sum of disparities

$$U_{\Sigma,\mathsf{max}} := \sum_{g \in \tilde{G}} U(g) = N \cdot \frac{R-1}{R}. \tag{6}$$

The similarity measure S-CLOP (**S**imilarity of multiple sets by **C**lustering with **LO**cal **P**runing) then is defined by

$$\text{S-CLOP}(G) := 1 - \frac{1}{U_{\Sigma,\mathsf{max}}} \sum_{g \in G} U(g) \ \in [0, 1] \tag{7}$$

and $\text{S-CLOP}(G^*) = \max_{g \in G} \text{S-CLOP}(G)$ defines the similarity of replicated LDA runs based on the identified optimal set of clusters G^*.

3.4 Using S-CLOP for LDAPrototype

As described in Section 3.1, the LDAPrototype algorithm (cf. Algorithm 1) relies on finding the medoid using a suitable similarity measure for LDA models. For this purpose, we use the S-CLOP measure from Section 3.3. In terms of the LDAPrototype selection procedure, to compute the S-CLOP similarities we always compare only two LDA runs, so that the clustering and the measuring of disparities aims at matching pairs of topics from the two different runs. Note that in this special case of comparing just two LDA runs with the same number of topics K, the normalization factor is $U_{\Sigma,\mathsf{max}} = 2 \cdot K \cdot \frac{1}{2} = K$. With

Table 2 Specifications of the six considered datasets

Dataset	Type	Time	M	V (Limit)	K	Source
reuters	Newspaper	1987	91	2141	5–15	[21]
economy	Wikinews	2004–2018	1855	7099 (5)	20	[13]
politics	Wikinews	2004–2009	4178	12 138 (5)	30	[13]
usatoday	Newspaper	06–11/2016	7453	25 486 (5)	50	[26]
tweets	Twitter	03–06/2020	3 706 740	17 208 (250)	25	[27]
nyt	Newspaper	1999–2019	1 993 182	74 218 (250)	100	[26]

respect to the definition of U we can simplify (7) to get

$$\text{S-CLOP}(G) = 1 - \frac{1}{2K} \sum_{g \in G} |g| \left(||g_{|1}| - 1| + ||g_{|2}| - 1| \right) \in [0, 1], \tag{8}$$

where $g_{|1}$ and $g_{|2}$ denote groups of g restricted to topics of the corresponding LDA run. By using the described pruning algorithm (cf. Section 3.3) in the two-LDA case, we obtain a set consisting of groups of size two or one. This means that a topic either has a directly similar counterpart topic in the other LDA, or that it forms a cluster itself. A topic always forms a cluster itself if it is either most similar to another topic from the same LDA, or if the complete linkage distance to one of the already found clusters of matched topics is smaller than to all remaining single topics of the other LDA run. Then, the medoid is determined by the run that maximizes the average pairwise S-CLOP value to all other LDA runs from the same set, namely $\bar{s}_i, i = 1, ..., R$ in Figure 1 and in Algorithm 1.

4 Data

In Section 6 we consider six different datasets, three of which are freely available through R packages [25]. Table 2 gives an overview of the datasets. Among them are three corpora consisting of traditional newspaper articles. The dataset *reuters* contains 91 articles from 1987 and is included in the package ldaPrototype [21]. The datasets *usatoday* and *nyt* are available to us via the paid service of [26], with more than 7000 articles and almost two million texts, respectively. They offer a good possibility to test the method on datasets of common and large size. For the latter, we consider all articles from the New York Times from 01/01/1999 to 12/31/2019. From the R package tosca [13], we also use the *economy* and *politics* datasets, which consist of nearly 2000 and just over 4000 Wikinews articles from 2004 – 2018 and 2004 – 2009, respectively. Also, in contrast to the other datasets, we consider a collection of nearly four million German-language tweets from March 19 – June 27, 2020, the first 101 days after then-German Chancellor Merkel's TV address on the coronavirus outbreak. For this purpose, 50 000 tweets with keywords related to the coronavirus were scraped every hour over the mentioned period using the Twitter API and duplicates were removed [27].

All six corpora are preprocessed in R with the packages tosca and tm [28], using common procedures in natural language processing (NLP). That is, duplicates from articles that occur more than once are removed, so that every unique article remains once. As an example, 204 articles were removed from the *usatoday* dataset, which previously contained 7 657 articles. As common in practice, characters are formatted to lowercase; numbers and punctuation are removed. In addition, a trusted stopword list [28] is applied to remove words that do not help in classifying texts in topics. Moreover, the texts are tokenized and words with a total count less or equal to a given limit are neglected. For example, for the *usatoday* dataset we choose this limit to be 5. This reduces the vocabulary size from 79 734 to $V = 25\,486$. For the larger datasets *tweets* and *nyt* we have set the limit higher to 250.

For all corpora, we heuristically choose a reasonable number of topics to model. We choose a higher number for larger and more general datasets. As can be seen in Section 6.4, for the *reuters* dataset we try different numbers of topics, namely $K = 5, \ldots, 15$, and investigate them with respect to the effects on modeling and runtime behavior. Other well known and widely used packages for preprocessing and/or modeling of text data are quanteda [29], topicmodels [30] and stm [31].

5 Study Design

In the following section, we apply the previously defined methods on the six presented datasets in different comparisons. For this purpose, the statistical programming software R 4.0.2 [25] and in particular the package ldaPrototype are used. This is based on an effective implementation in C/C++ of the LDA from the package lda [32]. For computation on a batch system or local parallelization, we use the packages batchtools [33] or parallelMap [34]. For modeling we always use the default parameters unless otherwise specified. For the number of topics K to be modeled, the methods deliberately do not provide a default. Our chosen parameters depending on the dataset are given in Table 2. The parameters α and η are chosen by default as `alpha` $=$ `eta` $= 1/K$, the Gibbs sampler runs for `num.iterations` $= 200$ iterations each time. For the computation using the thresholded version of the Jaccard coefficient tJacc defined in Equation (3), we choose `limit.abs` $= 10$, `atLeast` $= 0$, and, unless otherwise specified, `limit.rel` $= 0.002$. In addition, $R = 100$ runs of LDAs are modeled by default.

The general procedure is as follows: We select a prototypical LDA from a set of R LDAs using the presented method LDAPrototype. We repeat this procedure H times. Thus, we obtain the pairwise S-CLOP values of all combinations of R LDAs, H times each. The pairwise similarity of the H most representative LDAs, each selected from R LDAs, can then be evaluated using the S-CLOP measure. Then, the distributions of mean similarities of the simple replications $\bar{s_r}^{(h)}, r = 1, \ldots, R, h = 1, \ldots, H$ are compared to the distribution

$$\left.\begin{array}{c}\left\{\left(\mathrm{LDA}_r^{(1)}, \bar{s_r}^{(1)}\right) \mid r = 1, \ldots, R\right\} \\ \vdots \\ \left\{\left(\mathrm{LDA}_r^{(H)}, \bar{s_r}^{(H)}\right) \mid r = 1, \ldots, R\right\}\end{array}\right\} \quad \left\{\left(\mathrm{LDA}_{\mathrm{opt}}^{(h)}, \bar{s}_{\mathrm{opt}}^{(h)}\right) \mid h = 1, \ldots, H\right\}$$

Fig. 2 Sets of tuples consisting of LDA models and their mean LDA similarity values to all other LDAs from the same set. Schematic representation of the repetition strategy with H repetitions of the LDAPrototype method, each based on R basic LDAs. The H repetitions result in H prototypes (on the right), which in turn can be compared by pairwise model similarities.

of mean similarities of the prototypical LDAs $\bar{s}_{\mathrm{opt}}^{(h)}$. Figure 2 gives a representation of the sets of tuples consisting of LDAs and their mean similarity to the other LDAs from the same set. A location shift between the distributions represents an increase in reliability due to the use of the selection mechanism. We compute the reliability score rs for a set of LDAs based on their mean similarities $\bar{s_1}, \ldots, \bar{s_L}$ for a L that is equal to R or H as the arithmetic mean of the similarities (cf. Equation (1)), which can visually be interpreted as the area between $x = 0$ and the empirical cumulative distribution function (ecdf), naturally bounded on the y-axis by 0 and 1. Then, we are able to quantify the gain in reliability for a specific selection criterion comparing the reliability scores $rs\,(\bar{s_r}^{(h)} \mid r = 1, \ldots, R)$ for $h = 1, \ldots, H$ and $rs\,(\bar{s}_{\mathrm{opt}}^{(h)} \mid h = 1, \ldots, H)$. For a better (visual) comparability of ecdfs and scores, we use $H = R = 100$ in the following, unless otherwise stated.

Figure 3 shows a schematic diagram of the study design. The order corresponds to the subsections of Section 6. In Section 6.1, we first show a minimal example of the application of the S-CLOP measure to $R = 4$ runs. In doing so, we analyze the clustering behavior of the topics and exemplify the instability and thus limited reproducibility and reliability of the LDA results. We then show in Section 6.2 the improvement in reliability on the same dataset (*usatoday*) in dependence of the parameter R, the number of candidate LDAs. In addition, we show that LDAPrototype outperforms perplexity and NPMI regarding the gain in reliability measured by rs defined in (1). This is followed by a comparison of the use of the presented Jaccard coefficient tJacc in (3) with the other similarity measures (cf. Section A) cosine (A5), Jensen-Shannon (A4) and Rank Biased Overlap (A6), in Section 6.3. For this, we also use the *usatoday* dataset and also compare the similarity measures in terms of their runtime and parallelizability. Then in Section 6.4, on a smaller dataset (*reuters*), we compare the runtime and reliability gains for different numbers of topics K and different numbers of modeled LDA runs R. Finally, we analyze all six datasets in Section 6.5 and show that the method yields an increase in reliability regardless of the dataset and at the same time remains computable for large datasets.

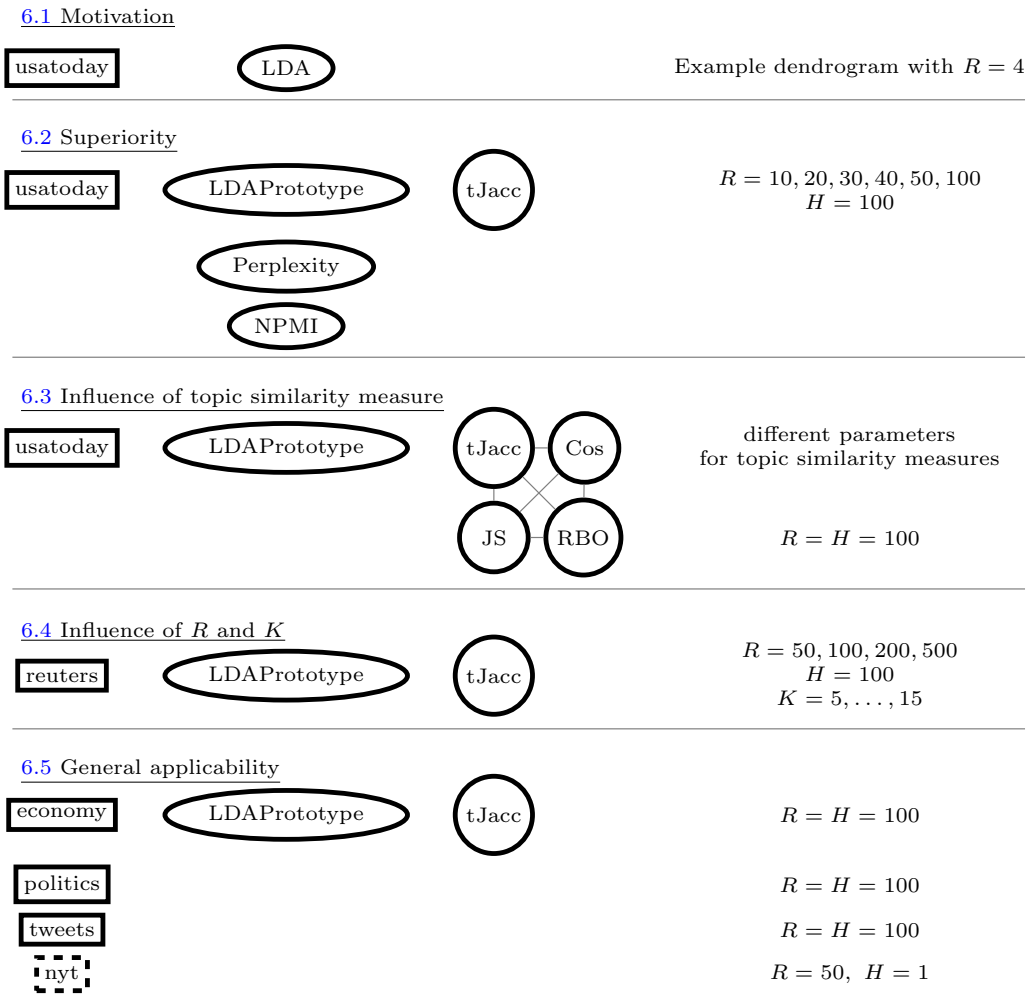


Fig. 3 Study overview: Illustration of the datasets, comparative methods, topic similarity measures, and parameters compared.

6 Results

Table 3 shows the runtimes for determining the LDAPrototype depending on the different datasets. Due to the size of the *nyt* dataset, only one prototype was determined, and only on $R = 50$ modeled LDAs. In addition, the calculation was not parallelized due to the required memory, so that it lasts around 130 days. For all other datasets $R \times H = 100 \times 100$ LDAs, and thus $H = 100$ LDAPrototypes were calculated. The runtimes are observed under parallelization on 4 cores and range from less than a minute for the *reuters* dataset to just over a day for the *tweets* dataset.

In Figure 4, the reliability gain of results from the LDAPrototype method can be seen in comparison to the reliability of basic LDA replications. The (red) curves, indicating the ecdfs of the mean pairwise similarities of the LDAPrototypes, differ significantly from the (black) ecdfs corresponding to the mean pairwise similarities of the simple LDA replications.

Table 3 Runtimes for determining the LDAPrototype on the six different datasets

Dataset	M	R	K	Cores	Min.	Mean	Max.	Unit
reuters	91	100	5	4	43.38	43.81	45.30	secs
economy	1855	100	20	4	8.25	8.33	8.69	mins
politics	4178	100	30	4	27.21	27.32	27.76	mins
usatoday	7453	100	50	4	3.33	3.42	3.56	hours
tweets	3 706 740	100	25	4	28.15	28.64	30.67	hours
nyt	1 993 182	50	100	1	-	130	-	days

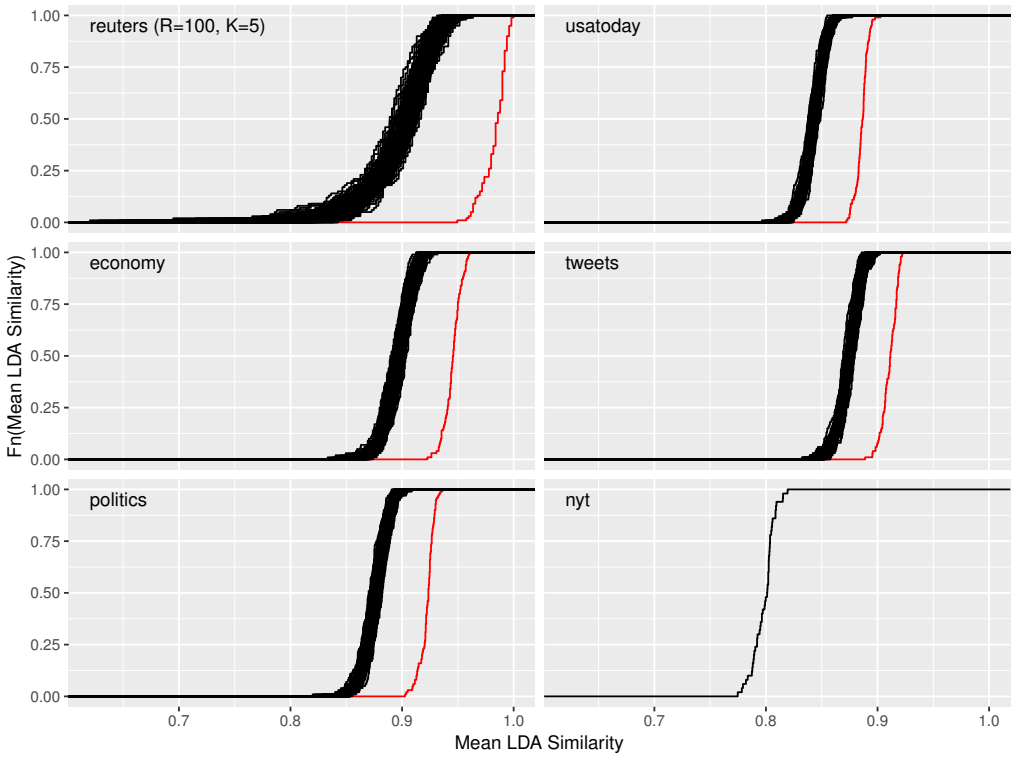


Fig. 4 Increase of reliability for five of the six different datasets and reliability of a single LDAPrototype for the *nyt* dataset; black: empirical cumulative distribution functions (ecdf) of the mean pairwise LDA similarites, red: ecdf of the mean pairwise LDAPrototype similarities

6.1 Cluster analysis and similarity calculation

In this section, we present an example analysis of a clustering result of four independent LDA runs based on newspaper articles from USA Today. We run the basic LDA four times, such that the number of runs to be compared is $R = 4$ and the total number of topics to be clustered is $N = R{\cdot}K = 4{\cdot}50 = 200$. To demonstrate how dissimilar replicated LDA runs can be, we cluster the $N = 200$ topics from the $R = 4$ independent runs with $K = 50$ topics each using the tJacc coefficient from Equation (3), complete linkage and the new introduced algorithm for pruning. The four runs were selected from 10 000 total runs. In fact, the runs *Run1* and *Run2* were chosen as the top two models

in mean similarity of the 100 prototypes, which means their points lie at the top of the very right (red) curve in the plot from *usatoday* in Figure 4. Their similarity values are 0.902 and 0.898 in the set of prototypes or 0.877 and 0.871 in the original sets, respectively. The model *Run3* was chosen as the worst of the 100 prototype models with a similarity value of 0.872 in the set of prototypes and 0.863 in its original set. *Run4* was chosen randomly as one of the worst models realizing a mean similarity to all other models in its original set of 0.807.

We apply hierarchical clustering with complete linkage to the 200 topics. The topics are labeled with meaningful titles (words or phrases). These labels were obtained by hand, based on the ranked list of the 20 most important words per topic. For this, the importance of a word $v = 1, \ldots, V$ in topic $k = 1, \ldots, K$ [32] is calculated by

$$I(v,k) = \frac{n_k^{(\bullet v)}}{n_k^{(\bullet\bullet)}} \left[\log\left(\frac{n_k^{(\bullet v)}}{n_k^{(\bullet\bullet)}} + \varepsilon \right) - \frac{1}{K} \sum_{l=1}^{K} \log\left(\frac{n_l^{(\bullet v)}}{n_l^{(\bullet\bullet)}} + \varepsilon \right) \right], \tag{9}$$

where ε is a small constant value which ensures numerical computability, which we choose as $\varepsilon = 10^{-5}$. The importance measure is intuitive, because it gives high scores to words which occur often in the present topic, but less often in average in all other topics.

Figure 5 shows a snapshot of the two dendrograms visualizing the result of clustering the 200 topics and of Algorithm 3 for clustering with local pruning. The vertical axes describe the complete linkage distance based on our tJacc coefficient with `limit.rel` = 0.002. In the lower dendrogram, the topic labels are colored with respect to the LDA run (*Run1*: grey, *Run2*: green, *Run3*: orange, *Run4*: purple). In the upper dendrogram, topic labels are colored according to the clusters obtained with our proposed pruning algorithm. In addition, every topic label is prefixed by its run number.

Looking at the upper dendrogram, we see that there are a number of clusters with identical labels for topics. This demonstrates that the four LDA runs produce a number of similar topics that are represented by similar word distributions. Examples for such stable topics are *Trump vs Clinton Campaign* colored yellow and *Olympics Medals* colored green.

However, there are also considerable differences visible. In the lower dendrogram in Figure 5, strikingly, there are several topics from *Run4*, highlighted in red, where no other topic is within a small distance. It is remarkable that *Run4* creates such a great number of individual topics, e.g. *Video Games*, *Gender Debate*, *TV Sports* (which includes words for describing television schedules of sport events) and *Terrorism*. Also, *Run4* leads to six explicit stopword topics, which is the maximum number compared to the other runs with four to six stopword topics.

In the upper dendrogram, the color depends on cluster membership. We measure combined stability of these four LDAs by applying the proposed

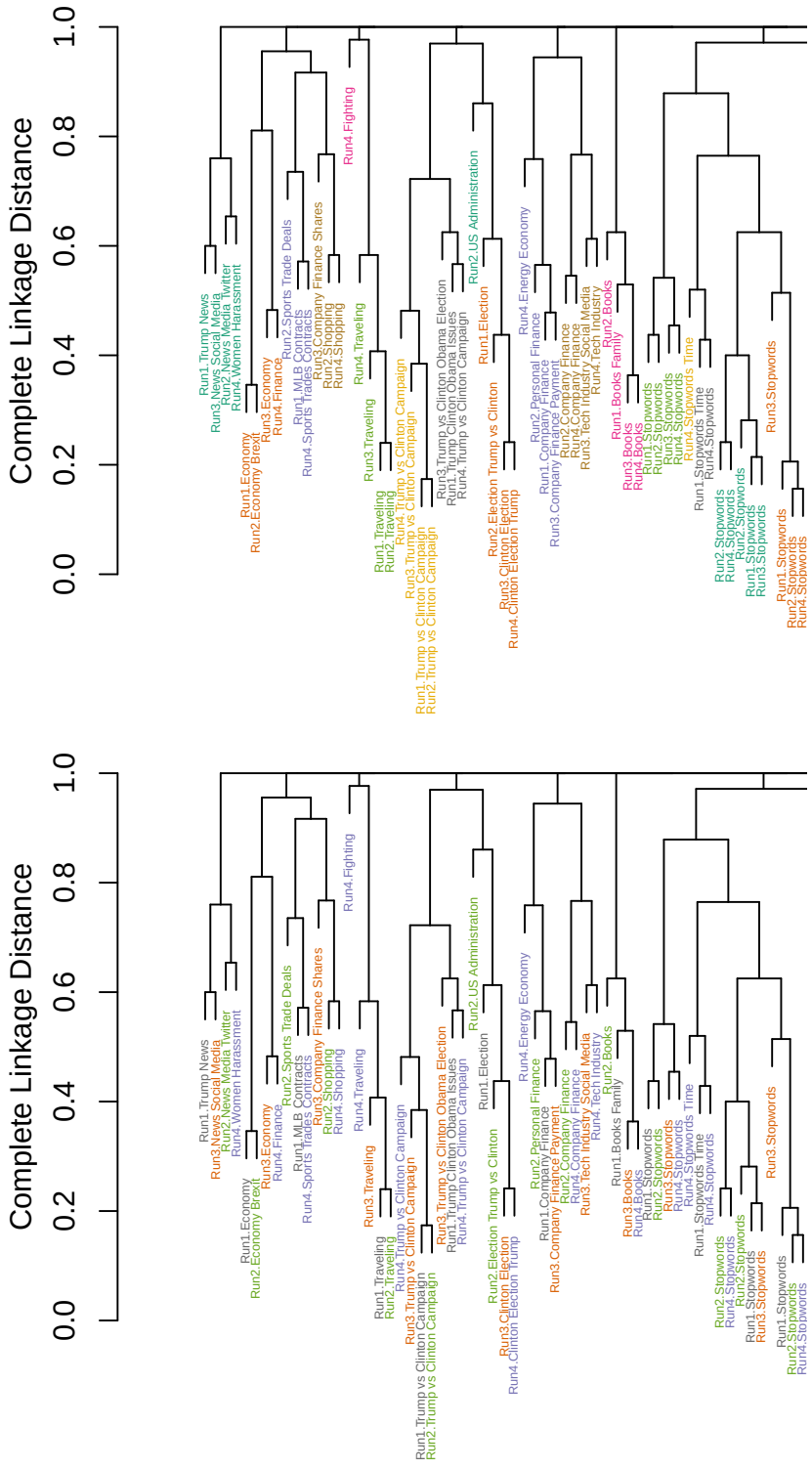


Fig. 5 Detail of dendrograms displaying 59 of the $N = 200$ topics from $R = 4$ selected LDA runs on the *usatoday* dataset with $K = 50$ topics each; bottom: colored by runs; top: colored by cluster membership. Complete dendrogram in Figure B1.

pruning algorithm (Algorithm 3) to the dendrogram. This leads to 61 clusters and a S-CLOP score of 0.83. The normalization factor is given by $U_{\Sigma,\max} = K \cdot (R-1) = 50 \cdot 3 = 150$, and the minimization of the sum of disparities yields $U^* = U_\Sigma(G^*) = 25$, resulting in S-CLOP $= 1 - 25/150 = 0.83$. There are seven single topics, one from each of the first three runs and four from *Run4*. The eleven clusters, which consist of exactly three topics, contain ten times a topic from *Run1*. Topics from *Run2* and *Run3* are represented nine times each, whereas only five of the mentioned clusters contain a topic from *Run4*. This shows that LDA run *Run4* strongly differs from the others. There are a lot of cases, where only one topic from this run is missing to obtain perfect topic clusters with exactly one topic from each run.

For comparison to the proposed local pruning algorithm, we once applied an established global criterion. Since 50 topics were originally modeled for each run, it is not reasonable to determine less than 50 clusters. Therefore, we try the global criterion with $50, \ldots, 70$ as the target cluster count. The largest similarity value according to S-CLOP based on the resulting clusters is obtained as 0.3 for 59 clusters. However, considering the dendrogram, we do not observe such large differences in the topic structures between the runs to justify such a low similarity. This shows the necessity of the presented local pruning method.

In addition, the dendrograms illustrate that random selection can lead to a poor model regarding interpretability and especially to some kind of an outlier model as *Run4*. This means that random selection can indeed lead to low reliability.

6.2 Increase of reliability

As an improvement for LDA result selection, we recommend to use the introduced approach based on prototyping of replications. It increases mean similarity, which comes along with an increase in reliability. We demonstrate how to determine a prototype LDA run as the most representative run out of a set of runs, based on our novel pruning algorithm. We show that this technique leads to systematically higher LDA similarities, which suggests a higher reliability of LDA findings from such a prototype run. This is indeed the case in comparison to basic LDA replications, but also in comparison to the well known selection criteria perplexity and NPMI. We calculate perplexity [30] by

$$\exp\left\{-\frac{\sum_{m=1}^{M}\sum_{v=1}^{V} n_\bullet^{(mv)} \log\left(\sum_{k=1}^{K} \hat{\theta}_{m,k}\hat{\phi}_{k,v}\right)}{\sum_{m=1}^{M} N^{(m)}}\right\}, n_\bullet^{(mv)} = \sum_{k=1}^{K} n_k^{(mv)} \quad (10)$$

and NPMI with respect to the definition from [16]. The authors offer a web service (see https://github.com/dice-group/Palmetto) to retrieve NPMI scores using the English Wikipedia as reference corpus. The score indicates a normalized pointwise mutual information for one topic using co-occurrences of the first 10 top words of the topic, determined using the score in (9). The NPMI

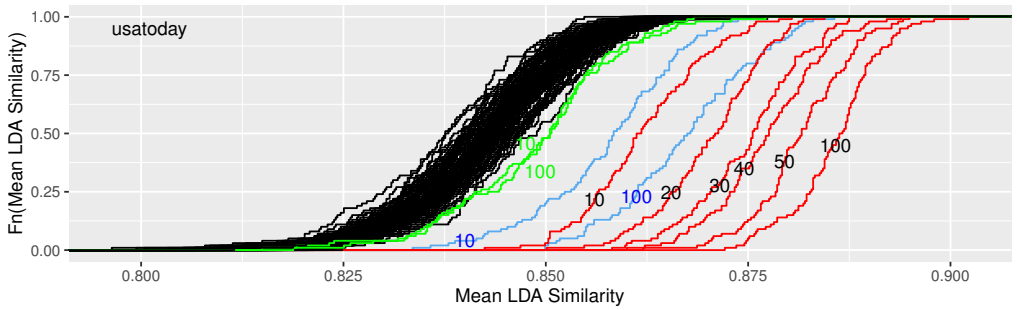


Fig. 6 Increase of reliability in dependence of the number of replications R on the *usatoday* dataset: ecdfs of the mean similarities calculated on $R = 100$ replications of randomly selected LDA runs (black) and on the $H = 100$ most representative prototype LDA runs (red) based on subsamples of $R = 10, 20, 30, 40, 50$ or all 100 LDA runs. For comparison, the ecdfs of the 100 selected LDA runs using perplexity (blue) and NPMI (green) based on the subsamples of 10 runs and all 100 runs are given.

score for one LDA model is then defined as the mean NPMI scores of all topics from the model.

The introduced similarity measure S-CLOP (7) quantifies pairwise similarity of two LDA runs by

$$1 - \frac{1}{50} \sum_{g \in G^*} U(g) = 1 - \frac{1}{100} \sum_{g \in G^*} |g| \left(||g_{|1}| - 1| + ||g_{|2}| - 1| \right),$$

where $K = 50$ is the number of topics per model and G^* an optimized set of topic clusters identified by our proposed pruning algorithm. We investigate the mean S-CLOP scores per LDA on the corpus from USA Today newspaper articles.

We propose to select the LDA run with highest mean pairwise similarity to all other runs. The following study shows that this is a suitable way to identify a stable prototype LDA, thus leading to improved reliability of LDA findings based on this particular run. We fit $R = 100$ LDA models and select the model with highest mean similarity as prototype. This procedure is repeated $H = 100$ times, which results in 100 prototype models. Then, for the 100 prototypes, also mean pairwise similarities to the other prototypes are calculated according to the schema in Figure 2. The results are visualized in Figure 6. The very right curve describes the empirical cumulative distribution function of the mean similarities obtained for the 100 prototypes. At the very left there are the $H = 100$ curves of the 100×100 original runs. In addition, we also determine 100 prototypes from subsamples. For this, only 10, 20, 30, 40 or 50 runs from each original set of $R = 100$ runs are randomly selected and are used for the following calculation steps. The resulting curves are plotted and labeled in Figure 6, the aggregated reliability scores are given in Table 4.

The minimum of the mean similarities from the original 100 sets of 100 models is 0.796, while the maximum is 0.877. It turns out that NPMI does not

Table 4 Reliability scores according to Figure 6. Comparison of the minimum, mean, and maximum reliability scores based on the $H = 100$ sets of basic LDA replications and the reliability scores after selection by LDAPrototype, perplexity, or NPMI in dependence of the number R of candidate models used.

R	10	20	30	40	50	100
Replications						
Min.	0.8338	0.8356	0.8378	0.8368	0.8359	0.8374
Mean	0.8429	0.8428	0.8426	0.8425	0.8425	0.8425
Max.	0.8515	0.8496	0.8483	0.8490	0.8487	0.8477
Selection						
LDAPrototype	**0.8622**	**0.8703**	**0.8750**	**0.8774**	**0.8816**	**0.8858**
Perplexity	0.8578	0.8612	0.8629	0.8629	0.8655	0.8674
NPMI	0.8485	0.8502	0.8519	0.8502	0.8493	0.8493

show a good ability to increase the reliability of LDA runs. For the reliability score of NPMI-selected runs, it makes little difference whether 10 or 100 models are available to the selection procedure. This suggests that the measure is better suited for other optimization goals than for increasing reliability. The perplexity does a relatively good job in the task of improving the reliability. For $R = 100$ candidate models, the relative reduction in the remaining improvement to the maximum reliability of 1 in comparison to random selection is 38% for our method LDAPrototype, namely $(1-0.8425)/(1-0.8858) = 0.1575/0.1142 = 1.38$, and 19% (0.1575/0.1326) for selection by perplexity. Therefore, the improvement in reliability for $R = 100$ is half that achieved by LDAPrototype. Overall, it can be seen that our method outperforms the other two popular and well-known selection methods in terms of increasing reliability.

Based on the findings from Figure 6 and Table 4, we recommend to fit at least 50 replications because this leads to an increase of similarity to 0.862 at the minimum and 0.895 at the maximum, or a reliability score (mean) of 0.8816, respectively. This corresponds to a relative reduction of the remaining possible improvement of 33%. Higher values for the number of repetitions are desirable. In general, the choice depends on the complexity of the corpus. Encapsulated topics or certain complicated dependency structures make the modeling procedure more prone to a larger span of possible fits and therefore to smaller mean similarity values. However, if computational power is limited, already taking the prototype model from 10 candidates considerably increases the reliability. Here, the minimum and maximum of mean similarity are 0.842 and 0.880 ($rs = 0.8622$), considerably higher values than these associated with random selection and 14% relative reduction of remaining improvement.

6.3 Comparison of the implemented similarity measures

Up to this point, we have performed all calculations with our tJacc (3) in Section 3.2. Now we study differences in reliability with different choices of measures of topic similarity. For this we compare cosine similarity (A5),

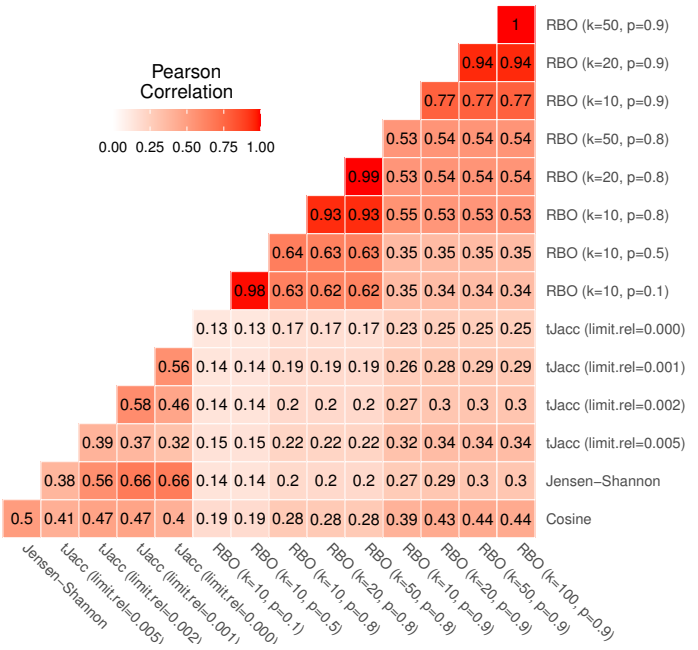


Fig. 7 Correlation matrix of pairwise LDA similarity values calculated using the four similarity measures with selected parameter combinations for 5000 topics on the *usatoday* dataset

Jensen-Shannon similarity (A4) and Rank Biased Overlap (RBO) (A6) with the tJacc coefficient. We consider, where applicable, effects of different parameter constellations on the correlation of the similarities. We show that the tJacc coefficient is a good choice for the topic similarity measure, because it improves reliability while not having a disproportionate runtime.

We again consider the *usatoday* dataset for the comparison. For the tJacc coefficient we set `limit.abs` $= 10$ and `atLeast` $= 0$ and vary `limit.rel` $\in \{0, 0.001, 0.002, 0.005\}$, for the RBO we choose eight combinations of $k \in \{10, 20, 50\}$ and $p \in \{0.1, 0.5, 0.8, 0.9\}$. We compute all pairwise topic similarities of the total $R \cdot K = 5000$ topics using the given measures and first consider the correlation of the pairwise LDA similarities based on these. In Figure 7 all pairwise correlations of the similarity values are given.

Thereby, clear patterns can be identified. The similarities obtained with cosine and Jensen-Shannon are correlated with a value of 0.50. The similarity values based on the tJacc coefficient correlate with the cosine similarity depending on the parameter in the range from 0.40 to 0.47. It is noticeable that the correlation initially increases from 0.41 to 0.47 when considering a longer tail of words belonging to a topic, but drops to 0.40 when considering the complete tail. In contrast, the correlation with Jensen-Shannon LDA similarities increases steadily from 0.38 for `limit.rel` $= 0.005$ to 0.66 when considering the complete tail, or these words that were assigned to that topic at least 11 times (`limit.abs` $= 10$). It is interesting to note that even the different versions of the tJacc coefficient are mostly less correlated with each other than the tJacc LDA similarities are with the Jensen-Shannon LDA similarities. This

Table 5 Runtime and parallelizability comparison of the four similarity measures computing pairwise similarities of $R \cdot K = 5000$ topics on the *usatoday* dataset; all runtimes refer to hours on 4 cores, a measure is better implemented in parallel for a larger parallelizability score $\in [0.25, 1]$.

	Measure k	Cos -	tJacc -	JS -	RBO 10	RBO 20	RBO 50	RBO 100
Parallel (4 Cores)	Min.	0.26	0.65	1.33	4.54	9.09	22.47	47.15
	Mean	0.29	0.68	1.39	5.05	11.40	24.75	51.62
	Max.	0.31	0.76	1.48	6.07	12.13	29.37	59.05
Serial	Time	0.86	2.49	4.58	-	27.26	-	-
Parallelizability	Score	0.75	0.93	0.83	-	0.60	-	-

illustrates that there can be significant differences between different choices for the threshold based on `limit.rel`.

As mentioned in Section 3.2, choosing the parameter `limit.rel` as 0.002 determines around 100 words per topic as relevant. Thus, one could assume that the corresponding similarities should correlate strongly with those of the RBO with $k = 100$. In fact, we see that the RBO is rather weakly correlated with the other three measures overall; in particular, for low values of $k \leq 50$ and $p \leq 0.8$. The corresponding correlations are all below 0.30. However, it is also noticeable that for increasing k and p, the RBO similarities appear to be increasingly correlated with the other three similarity measures. This is due to the fact that for larger values of k and p the RBO converges to a measure very similar to the Jaccard coefficient or for $p \neq 1$ to the AverageJaccard defined in Equation (A1) in Appendix A. This means that RBO similarities calculated with parameters $k = 100$ and $p = 1$ should be very close to tJacc similarities with `limit.rel` $= 0.002$. In the present case, however, even with 0.9, p is still much smaller than 1. Note that $0.9^{100} \approx 0$ and thus the 100th word in the ranked list gets practically no weight, which is exactly the idea of the RBO. Therefore it is not recommended to choose the parameter p close to 1. Alternatively, a better approach would be to choose an implementation based on Jaccard, e.g. AverageJaccard, because it is faster to implement. These findings show that k and p should always be chosen dependent on each other. One can also see in Figure 7 that words from rank 50 onwards no longer have a significant influence with a choice of $p = 0.9$. The correlation between the LDA similarities based on RBO with $k = 50, p = 0.9$ and $k = 100, p = 0.9$ is 1.

The four measures considered have different complexities in terms of their implementations. While the cosine similarity can be computed very quickly, the tJacc coefficient and the Jensen-Shannon similarity require computationally intensive precomputations, which increases the runtime. Cosine similarity and Jensen-Shannon similarity have no parameters to be set. For the tJacc coefficient, the calculations do not depend on the chosen parameters. The calculation of the RBO is generally time-consuming, since values must be calculated individually for all considered depths up to the maximum rank, which

also means that the runtime for the calculation of the RBO strongly depends on the parameter k.

The runtimes are documented in Table 5. The times here are in hours, the calculations were run on four cores in parallel and are based on at least 100 different values. For each measure, a serial calculation was performed once in each case in order to be able to give a value for estimating the parallelizability of the calculations or implementations. It can be seen that the cosine similarity is the fastest to calculate with well under one hour. The thresholded Jaccard coefficient requires slightly more than twice as much time, while the Jensen-Shannon similarity computes with 83 and 275 minutes, respectively, again almost twice as long as the tJacc coefficient. The Rank Biased Overlap with $k = 100$ takes even over 2 days in parallel computation. This is also due to the fact that the implementation is only parallelized with a score of 0.6, i.e. one achieves only 60% of the maximum possible time saving through parallelization. With a maximum score of 1, the calculation would only take 30.97 hours instead of 51.62. The cosine similarity is also not implemented maximally parallelized. This is due to the overall low runtime. The tJacc coefficient, instead, is implemented in parallel with an approximate maximum time saving of 0.93.

After comparing the measures in terms of runtime and correlation of the resulting pairwise S-CLOP values, we restrict the analysis to one parameter combination per measure and compare the increase in reliability in dependence of the measure. In Figure 8, these are plotted against each other. We compare the cosine similarity, Jensen-Shannon similarity, the tJacc coefficient with `limit.rel = 0.002` and the RBO with $k = 20, p = 0.9$. On the diagonal of the plot matrix, the known ecdfs of the LDA similarities are shown in black and the ecdf of the LDAPrototypes in red, respectively. In addition, we calculated - using the same measure - the similarities of the LDAPrototypes, that were determined based on the other three topic similarities. The colors of the ecdfs correspond to the color of the box of similarity measures. This allows the different measures to be analyzed in terms of their selection behavior taking into account the level differences of the similarity values. Table 6 gives the corresponding reliability scores of the ecdfs on the diagonal. The lower triangular plot matrix provides the LDA similarities as pairwise correlation plots, the upper matrix the corresponding correlations themselves. Determining the pairwise LDA similarities based on 100 experiments with $R = 100$ replications each yields $100 \cdot (R-1)R/2 = 495\,000$ values per similarity measure, on which the heatmaps and correlations are both based.

It can be seen that the LDA similarities according to the tJacc coefficient and according to the Jensen-Shannon similarity are most highly correlated with each other. The point cloud is least circular, but rather narrow and elliptical in shape. It can also be seen that the RBO LDA similarities are very different from the others. In the lower right plot (outlined in blue), the three ecdfs of different colors show significant differences from the red curve, with the cosine curve showing even slightly more similarity than the curves of the other two measures. Similarly, the blue curve is in the Jensen-Shannon (purple)

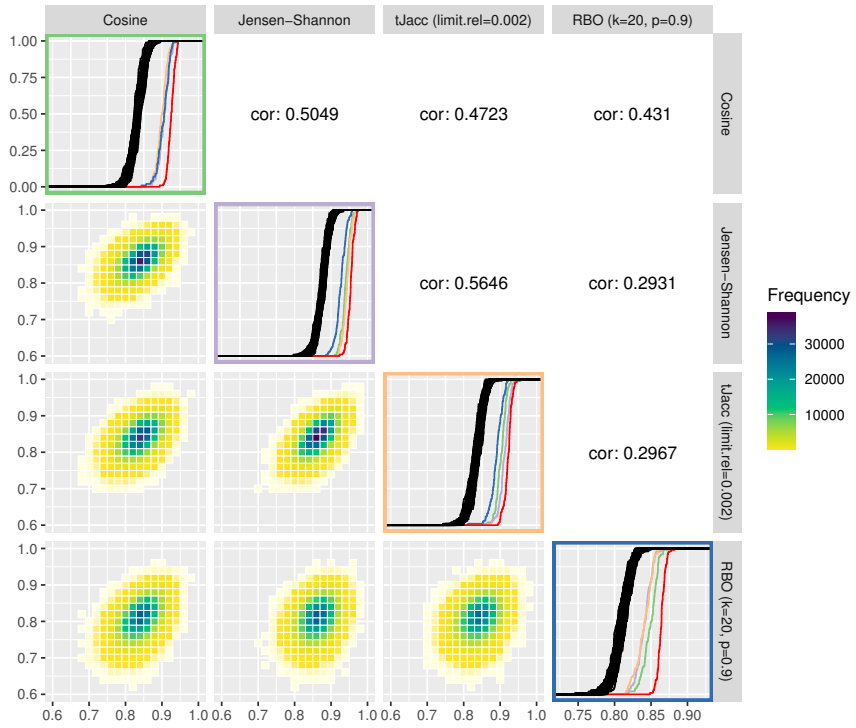


Fig. 8 Comparison of four selected similarity measures regarding their increase of the reliability of LDA results: on the diagonal the ecdfs of the LDA similarities based on the different similarity measures are shown in black, in addition the similarities of the LDAPrototypes are shown in red or in further colors for the LDAPrototypes determined on the basis of the other measures (green for cosine, purple for Jensen-Shannon, orange for tJacc and blue for RBO); below the diagonals correlation plots of the LDA similarities according to the similarity measure are shown as heatmap, above the diagonals the corresponding correlations are given.

and tJacc (orange) windows far behind the other colored ecdfs. The cosine LDA similarities (top left, green) differ barely for all the foreign-determined LDA prototypes, but the red curve sets itself apart from the others. Thus, the selection by cosine similarity obviously also differs significantly from the others.

The plots from Figure 8 suggest that all four measures differ with regard to their selection criteria. However, a qualitative ranking which of the measures increases the reliability of the results the most is difficult because for this question the true evaluation measure has to be identified first, which is a vicious circle. For this reason, all these measures have their justification to be used within the procedure. We prefer to use the thresholded Jaccard (tJacc) coefficient because it has plausible heuristics and reasonable runtime. Its selection of the LDAPrototype strongly correlates with that of the Jensen-Shannon similarity, for which, according to Appendix A, besides the tJacc coefficient itself, the best results in terms of correlations with human perceptions could be obtained.

Table 6 Reliability scores according to Figure 8. Comparison of the topic similarity measures with respect to the increase in reliability. Each column shows the reliability measures depending on the used topic similarity measure, the lower four rows indicate which similarity measure is used for the selection of the $H = 100$ prototypes based on the respective $R = 100$ basic LDAs.

	Comparison of LDA runs based on			
	Cos	tJacc	JS	RBO
Replications				
Min.	0.8364	0.8374	0.8587	0.8039
Mean	0.8413	0.8425	0.8634	0.8101
Max.	0.8459	0.8477	0.8666	0.8155
LDAPrototype				
Cos	**0.8878**	0.8764	0.8951	0.8487
tJacc	0.8748	**0.8859**	0.8970	0.8403
JS	0.8778	0.8811	**0.9019**	0.8410
RBO	0.8771	0.8700	0.8884	**0.8642**

6.4 Comparison of different values for the parameters R and K

In addition to the choice of the similarity measure, the choice of the parameters K, number of topics to be modeled, and R, number of replications, also has a large influence on the runtime of the method. In Section 6.2 it has already been shown that larger values for R result in a larger increase in reliability. We will confirm these findings based on the *reuters* dataset on the one hand and on the other hand bring them into a combined comparison with the number of modeled topics.

In Table 7 the runtimes for the determination of the LDAPrototypes based on the calculations of the topic similarities by the tJacc coefficient are given. Obviously, the runtime increases more than linear in both in the number of topics K to be modeled and in the number of replications R. This is due to the fact that the computation of the matrices of topic similarities have a quadratic complexity, since pairwise similarities are computed. The runtime for modeling the LDAs is linear in the parameters R and K.

Figure 9 shows the corresponding plots for the increase in reliability for all combinations of $R = 50, 100, 200, 500$ and $K = 5, \ldots, 15$ and Table B1 the corresponding reliability scores. Consistent with expectations, for fixed K, increasing the replication number from 50 to 500 does not change the location parameter for the ecdfs of the mean pairwise LDA similarities. However, the variance of the similarities decreases due to the higher replication number. It can be seen that for fixed R and increasing topic number K the level of similarities decreases. From an average similarity of 0.90 for $K = 5$ for the LDA replications, the value decreases to 0.75 for $K = 10$ and 0.65 for $K = 15$. The increase in reliability is marked by the red ecdfs and is clearly pronounced for all parameter combinations. The gain is larger for higher topic numbers, so the level of LDAPrototype similarity does not decrease quite as much with

Table 7 Runtime comparison of LDAPrototypes on the *reuters* dataset for choices of R and K; all runtimes refer to minutes on 4 cores.

R	50	100	200	500
$K = 5$				
Min.	0.25	0.72	2.46	16.01
Mean	0.26	0.73	2.49	16.31
Max.	0.31	0.76	2.77	17.16
$K = 10$				
Min.	0.47	1.43	5.47	36.45
Mean	0.48	1.47	5.57	37.03
Max.	0.53	1.64	5.84	38.25
$K = 15$				
Min.	0.79	2.57	9.81	69.28
Mean	0.80	2.65	10.03	70.03
Max.	0.82	2.81	10.37	71.58

increasing parameter K. The gain is also larger for increasing R. This is an expected behavior because then each prototype is determined from a larger set of individual LDAs and thus each LDAPrototype becomes even more reliable. For $K = 5$, the similarities of the prototypes thus increase from 0.97 for $R = 50$ to close to 1 for $R = 500$ (with LDA similarities around 0.90). For $K = 9$ they increase from 0.91 to 0.95 (0.78) and for $K = 14$ from 0.84 to 0.90 (0.67).

The findings from Figure 6 are confirmed here. With increasing R the reliability gain increases. However, already for small values of R a clear increase is recognizable. Accordingly, R should be chosen as large as possible depending on the available computing power.

6.5 Comparison of the introduced datasets

For the corpus of 7453 newspaper articles from the USA Today from 01/06 to 11/30/2016 and the *reuters* dataset with $M = 91$, an increase in reliability could clearly be shown. Lastly, we show that this increase results independently of the dataset, so that the LDAPrototype method can be reasonably applied to many different types of text data.

Figure 10 and Table 8 in particular show the increase in reliability for the datasets *tweets*, *politics*, and *economy* that have not yet been considered so far. For comparison, the corresponding plots for the *usatoday* dataset are shown, for *reuters* with $K = 10$ and the four different values for $R = 50, 100, 200, 500$ (cf. Section 6.4). Figure 4 already showed the computability for large datasets on the example of the *nyt* dataset. We did not repeat the computation of the LDAPrototype 100 times for this dataset due to the comparatively higher runtime. The *tweets* dataset, with 3 706 740, does consist of a larger number of individual documents than the *nyt* dataset (1 993 182, cf. Table 2). However, due to the more specific topic structure, we chose a smaller topic number for this dataset. Together with the fact that the modeled tweets are much shorter than journalistic texts from the New York Times, this leads to a significantly

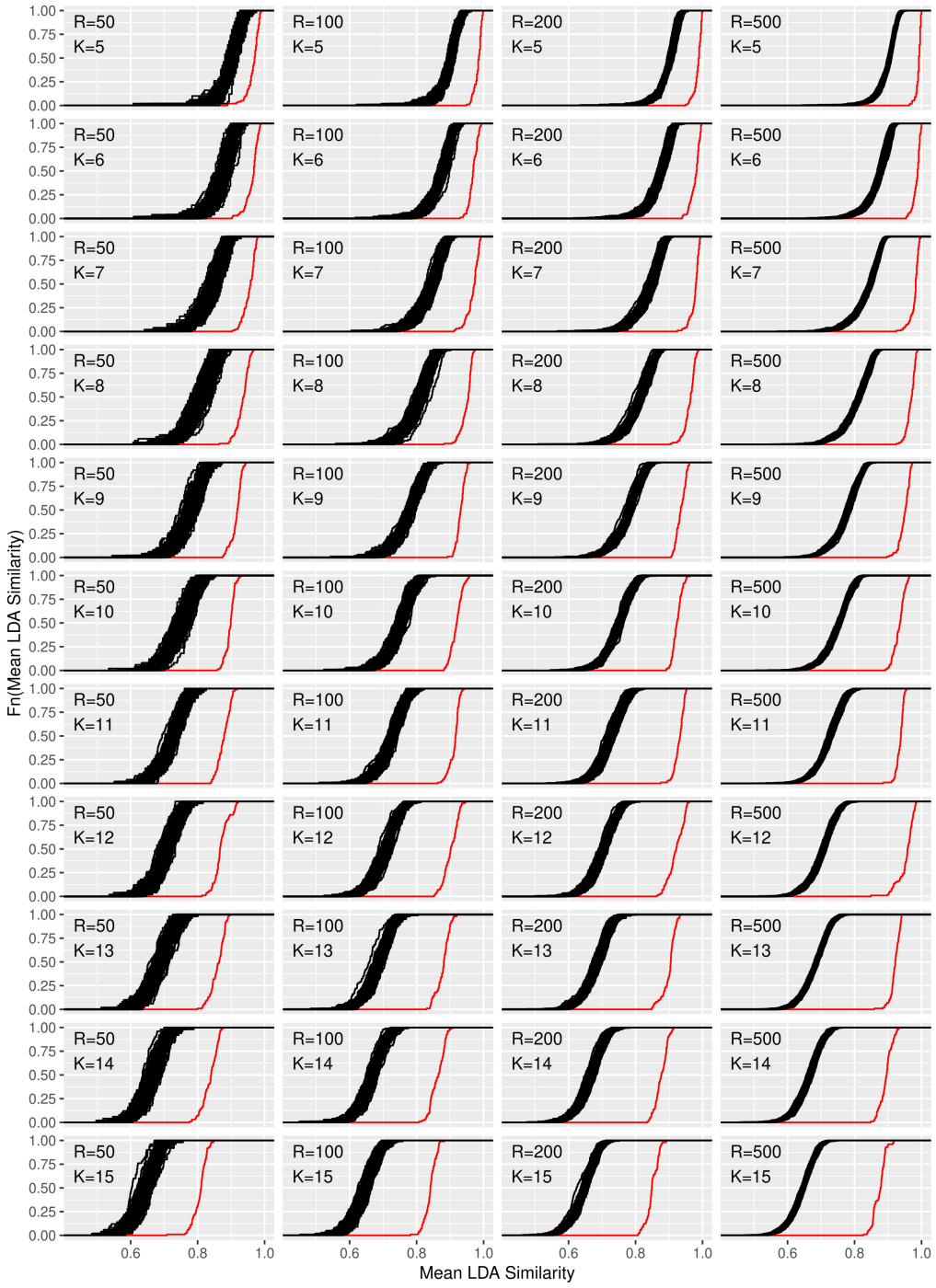


Fig. 9 Increase of reliability for the *reuters* dataset for $R = 50, 100, 200, 500$ and $K = 5, \ldots, 15$.

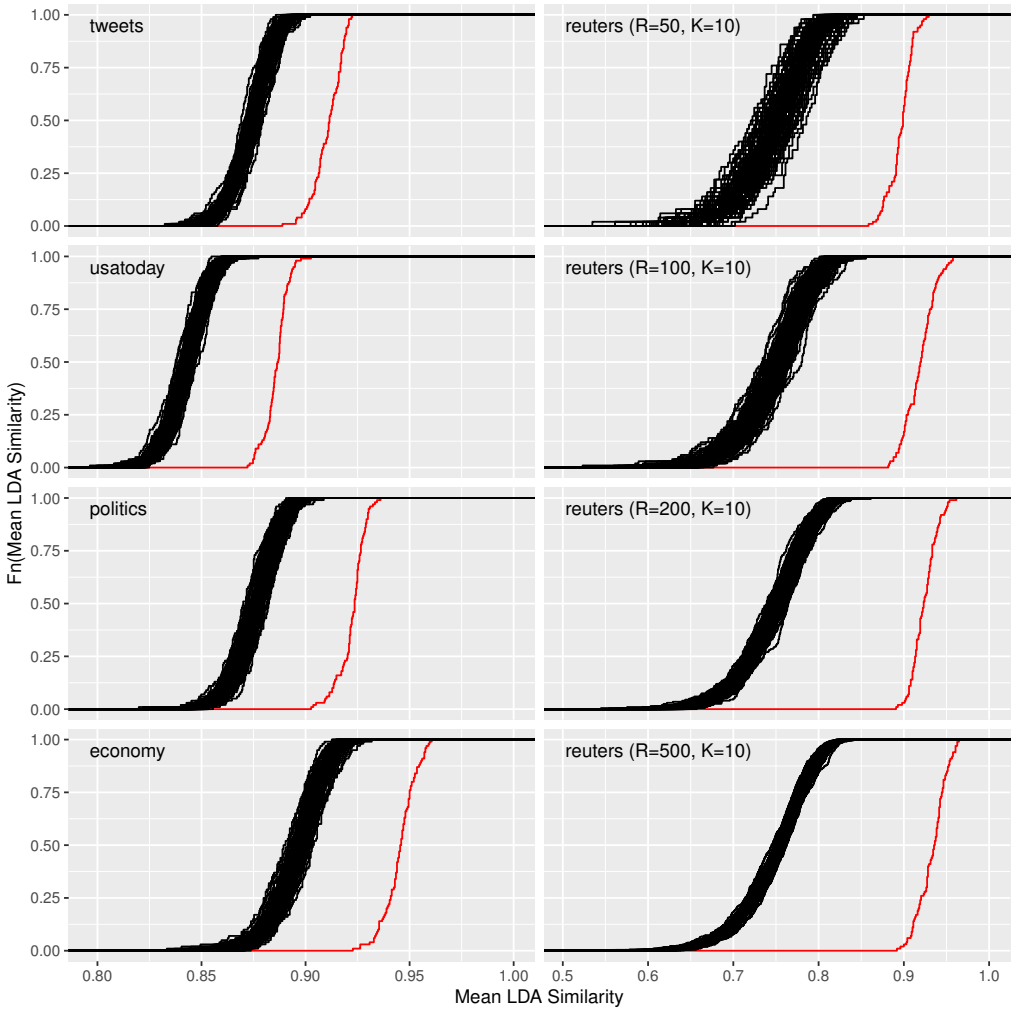


Fig. 10 Increase of reliability for the datasets *tweets*, *usatoday*, *politics*, and *economy* with $R = 100$ and for *reuters* with $R = 50, 100, 200, 500$

lower runtime of just over one day per LDAPrototype than for the *nyt* dataset with about 130 days (cf. Table 3).

The plots show a lower gain in reliability for *tweets* than for the dataset *economy*, for example. This might surprise because the similarities of the basic LDA repelications are already somewhat higher for the latter. To be concrete, the similarities increase from about 0.88 to 0.94 for *economy* and from 0.87 to 0.91 for the dataset consisting of tweets. This reduced gain could be due to the shorter texts and the resulting higher uncertainty in the modeling. As a result, the instability of the LDA on this dataset is more pronounced, so that higher reliability increases are only possible with larger values for R. On the right side of Figure 10 the results from the Sections 6.2 and 6.4 are recapitulated. The increase of reliability increases steadily with an increasing number of LDA replications R.

Table 8 Reliability scores according to Figure 4 and 10. Comparison of the increase in reliability obtained by using LDAPrototype on the datasets tweets, politics and economy ($R = H = 100$). In addition, the reliability score of the basic LDA replications for the nyt dataset is given ($R = 50$).

	tweets	politics	economy	nyt
Replications				
Min.	0.8683	0.8711	0.8885	-
Mean	0.8735	0.8759	0.8963	0.7982
Max.	0.8790	0.8809	0.9029	-
Selection				
LDAPrototype	0.9108	0.9224	0.9453	-

7 Discussion

Topic modeling is popular for understanding text data, though the analysis of the reliability of topic models is rarely part of applications. This is caused by plenty of possibilities for measuring stability, but missing strategies for increasing reliability without touching the original fitting procedure.

We have presented a novel method to address the reliability issue caused by the inherent instability of the LDA procedure. For this purpose, we want to improve the reproducibility of the results and we talk about highly reliable results if they can be reproduced very well. The presented method is based on the idea of modeling a set of LDAs and then selecting the best, in this case the most central, model through a selection mechanism. We call this medoid of several LDA runs LDAPrototype. We deliberately choose not the best-fit model according to one of the well-known - mostly likelihood-based - measures [8, 30], but the LDA that agrees most with all other LDAs obtained from the same text data using the same parameter settings. We were able to show that the selection of the LDA using our LDAPrototype method strongly increases the reliability. In comparison, for the datasets under consideration in this paper, the improvement turns out to be about twice as large as that obtained from a selection based on perplexity. Moreover, model selection with the optimal NPMI does not significantly improve the reliability compared to the basic LDAs, regardless of the number of candidate models.

In various analyses, we have shown that the presented method increases the reliability of the results. By applying it to different datasets, we have first shown its feasibility due to the implemented R package and at the same time that it produces the desired increase in reliability irrespective of the dataset. We also investigated the influence of the parameters of the number of modeled topics K and number of LDA replications R on this increase. Furthermore, we presented differences in the determination of the LDAPrototype based on the presented similarity measures. All four measures under consideration resulted in an increase in reliability. No clear ranking could be obtained. The decision for the thresholded Jaccard (tJacc) coefficient as the default measure is based on the combination of visible increase in reliability, interpretability of the measure,

fast implementation, and supporting arguments from studies [35, 36] regarding correlation with human perception.

One limitation of the method is that the repetition strategy could be computationally demanding on large datasets with a large number of topics. This can be seen in the example computation on the New York Times dataset (nyt). However, the results on larger datasets are naturally more stable due to the large size of the dataset. In fact, the study showed that LDAPrototype basically has a larger effect in terms of increasing the reliability of the results on small datasets. For this reason, we see the main application of the method in scenarios with small to medium sized datasets, where computability - also through parallelization and easy interfacing to high performance clusters - is thus not a concern.

The quality of the results of LDAPrototype in terms of correlation with human perception was not in the scope of the paper, but we plan to investigate it in further studies. For this, a study with human coders is necessary and different models like the basic LDA, LDAPrototype, Structural Topic Model, perplexity or NPMI optimized and potentially other models are evaluated regarding their quality by human coders. For this purpose, we focus on meaningful and distinct topics.

In several application examples - for example in communication studies - our proposed method has already produced well interpretable topics in an automated way [e.g. 37]. A side effect from our method is that LDAPrototype prevents a human "selection algorithm" that chooses the LDA model that best supports the hypothesis. The LDAPrototype methodology also forms the basis for a new LDA method RollingLDA [38], which was developed for the construction of time-consistent time series using LDA.

The presented idea of selecting a prototypical LDA from a set of LDA runs can be transferred to other topic models as well. For example, the Structural Topic Model offers not only pre-initialized topics, but also the possibility of random initialization. This causes the issue of limited reliability of interpretations due to the lack of reproducibility of the results. At this point, the reliability may also increase using an analogous procedure to the method LDAPrototype.

Acknowledgments. The present study is part of a project of the Dortmund Center for data-based Media Analysis (DoCMA). In addition, the authors gratefully acknowledge the computing time provided on the Linux HPC cluster at TU Dortmund University (LiDO3), partially funded in the course of the Large-Scale Equipment Initiative by the German Research Foundation (DFG) as project 271512359.

Declarations

Competing interests. The authors have no competing interests to declare that are relevant to the content of this article.

Code and data availability. The methods themselves are published in the R package ldaPrototype both via CRAN and as a devoloping package on

GitHub. Four of the six datasets used are freely available. The other two datasets are unfortunately not publishable for licensing reasons. All analysis scripts will be made available in an associated GitHub repository.

Appendix A Additionally implemented similarity measures for topics

For our selection procedure LDAPrototype, we choose the tJacc coefficient from Section 3.2 as main similarity measure for comparing topics based on their word count vectors. In the literature, several alternatives are discussed, from which we compare the three most promising to our thresholded version of the Jaccard coefficient in Section 6.3.

While [10] determine topic similarity with a Jaccard coefficient of the top 9 words per topic across multiple runs and measure stability with the median of the topic similarities, [18] use the cosine similarity (see below). For repetitions of the same modeling procedure they match topics with the highest cosine similarity, which additionally has to be greater than an arbitrarily selected threshold 0.7. Then, for two models the similarity is calculated as the share of topic matches, and for more than two models by the mean of all pairwise shares.

[39] and [40] determine topic similarities with an average Jaccard coefficient

$$\text{AverageJaccard}(A, B) = \frac{1}{N} \sum_{n=1}^{N} \frac{|A_n \cap B_n|}{|A_n \cup B_n|}, \tag{A1}$$

where A_n and B_n define the sets of the first n words of the ordered lists from the word sets A and B. They choose $N = 5$ and find the best matching topics of different LDA runs based on this measure with the hungarian method [41]. The authors try to encounter the problem that more than two runs of topics have to be matched by learning a reference model. They calculate the similarity of one LDA to the reference LDA as the mean average Jaccard coefficient over all matched topics, characterized by their word sets Z_{k^*} to the topics $Z_k^{(\text{ref})}$ of the reference model. Here Z_{k^*} denotes the reordered topics' word lists Z_k of the LDA run, so that matched topics have the same index. Analogously, they calculate stability over a number of R replications as the mean over the pairwise similarities against the topics' word lists of the (predetermined) reference model $Z_k^{(\text{ref})}$ by

$$\frac{1}{R} \sum_{r=1}^{R} \left(\frac{1}{K} \sum_{k=1}^{K} \text{AverageJaccard} \left(Z_k^{(\text{ref})}, Z_{k^*}^{(r)} \right) \right). \tag{A2}$$

One drawback of this approach is the specification of the reference model, which should be a good representative of all other LDAs. It is non-trivial to

determine this representative model. Therefore, our approach follows an opposite strategy. We first calculate similarities between models and then determine the prototype model, i.e. the most representative LDA run, based on these values.

[35] argue that Jensen-Shannon divergence [42] is one of the best similarity measures based on word distributions considering correlation with human judgments. It is a symmetric version of the Kullback-Leibler divergence [43]

$$\text{KLD}\left(\boldsymbol{q}_1, \boldsymbol{q}_2\right) = \sum_{v=1}^{V} q_{1,v} \log \frac{q_{1,v}}{q_{2,v}}, \;\; \boldsymbol{q}_1, \boldsymbol{q}_2 \in (0,1]^V \tag{A3}$$

and treated as similarity measure defined as

$$\begin{aligned} \text{JS}(\boldsymbol{w}_i, \boldsymbol{w}_j) &= 1 - \left(\text{KLD}\left(\boldsymbol{p}_i, \frac{\boldsymbol{p}_i + \boldsymbol{p}_j}{2}\right) + \text{KLD}\left(\boldsymbol{p}_j, \frac{\boldsymbol{p}_i + \boldsymbol{p}_j}{2}\right)\right)/2 \\ &= 1 - \text{KLD}(\boldsymbol{p}_i, \boldsymbol{p}_i + \boldsymbol{p}_j)/2 - \text{KLD}(\boldsymbol{p}_j, \boldsymbol{p}_i + \boldsymbol{p}_j)/2 - \log(2), \\ \boldsymbol{p}_l &= (p_{l,1}, ..., p_{l,V}) = \left(n_l^{(\bullet 1)}, ..., n_l^{(\bullet V)}\right)/n_l^{(\bullet\bullet)}. \end{aligned} \tag{A4}$$

[44] use a normalized variant of the Jensen-Shannon divergence (they refer to it as the Kullback-Leibler divergence) to measure topic stability, and suggest using the measure to identify stable topics in repeated runs. In our study in Section 6.3, we compare the Jensen-Shannon divergence to other approaches for topic similarity measures. Moreover, [35] found out that a standard Jaccard coefficient is able to realize higher correlations to human judgments than other common similarity measures on specific datasets.

[36] showed that Jaccard coefficients perform on par with Jensen-Shannon divergence and outperform a number of other popular similarity measures like cosine similarity, which is defined as

$$\cos(\boldsymbol{w}_i, \boldsymbol{w}_j) = \frac{\sum_{v=1}^{V} n_i^{(\bullet v)} n_j^{(\bullet v)}}{\sqrt{\sum_{v=1}^{V} \left(n_i^{(\bullet v)}\right)^2} \sqrt{\sum_{v=1}^{V} \left(n_j^{(\bullet v)}\right)^2}}. \tag{A5}$$

and the mentioned Kullback-Leibler divergence. To quantify the quality of the similarity measures they compare the negative log-likelihood of the model as an indicator how well the model explains the data. They swap the best matching topics from models of two time slices and interpret an increase of the negative log-likelihood as deficiency of the specific similarity measure.

Another option for measuring topic similarity introduced by [45] is the Rank Biased Overlap (RBO) [46] for comparing ranked lists. The similarity is

defined as

$$\text{RBO}(A, B) = 2p^k \frac{|A_k \cap B_k|}{|A_k| + |B_k|} + \frac{1-p}{p} \sum_{d=1}^{k} 2p^d \frac{|A_d \cap B_d|}{|A_d| + |B_d|} \tag{A6}$$

with parameters $k \in \mathbb{N}$ setting the maximum depth of evaluation and $p \in (0, 1)$ controlling the influence of higher ranked words. While the measure seems to be useful because it implements a more flexible form of a Jaccard coefficient, the authors do not investigate stability of LDA models based on RBO. Moreover, the calculation of the measure is very time consuming.

For our analysis we prefer the thresholded version of the Jaccard coefficient tJacc as defined in Equation (3), because in Section 6.3 we show that it performs on par with the others. In addition, it is calculated faster than the Jensen-Shannon divergence and especially than the RBO. In comparison to the cosine similarity, calculation of the tJacc is computationally more demanding, but may lead to an increased interpretability. In view of the large computation demand for the LDA modeling in big data scenarios, this runtime increase is acceptable and rather minor. Our R package ldaPrototype offers the possibility of calculating the prototype based on topic similarities calculated using the cosine, tJacc, Jensen-Shannon or RBO similarity measure.

Appendix B Additional Figures and Tables

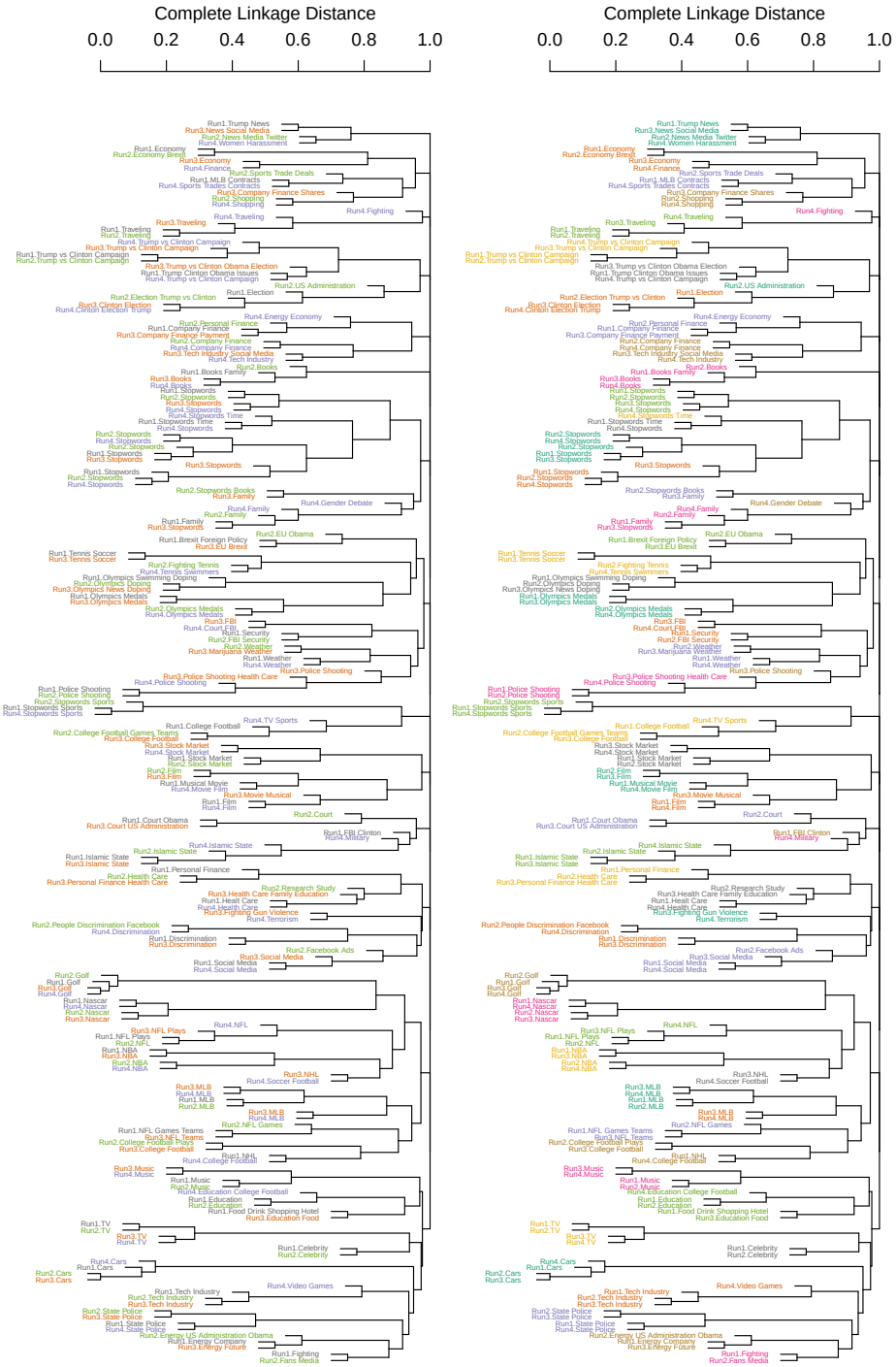


Fig. B1 Dendrograms of $N = 200$ topics from $R = 4$ selected LDA runs on the usatoday dataset with $K = 50$ topics each; left: colored by runs; right: colored by cluster membership. Detail can be found in Figure 5.

38 *LDAPrototype: Improving Reliability of LDA*

Table B1 Reliability scores according to Figure 9. Comparison of different choices of R and K on the reuters dataset for the increase in reliability obtained by using LDAPrototype.

	Replications			Selection
K	Min.	Mean	Max.	LDAPrototype
$R = 50$				
5	0.8754	0.8988	0.9205	0.9657
6	0.8456	0.8696	0.9026	0.9626
7	0.8091	0.8394	0.8623	0.9542
8	0.7763	0.8043	0.8370	0.9340
9	0.7397	0.7740	0.8000	0.9157
10	0.7167	0.7473	0.7806	0.8967
11	0.7000	0.7220	0.7466	0.8784
12	0.6798	0.7029	0.7288	0.8707
13	0.6615	0.6844	0.7116	0.8596
14	0.6321	0.6625	0.6900	0.8369
15	0.6045	0.6415	0.6639	0.8066
$R = 100$				
5	0.8806	0.8975	0.9098	0.9834
6	0.8521	0.8689	0.8933	0.9688
7	0.8123	0.8376	0.8541	0.9681
8	0.7832	0.8035	0.8362	0.9490
9	0.7572	0.7752	0.7952	0.9313
10	0.7279	0.7479	0.7660	0.9189
11	0.7022	0.7224	0.7430	0.9129
12	0.6815	0.7019	0.7240	0.9018
13	0.6543	0.6832	0.7014	0.8788
14	0.6439	0.6647	0.6849	0.8626
15	0.6233	0.6436	0.6641	0.8407
$R = 200$				
5	0.8878	0.8975	0.9067	0.9861
6	0.8565	0.8684	0.8832	0.9780
7	0.8189	0.8372	0.8544	0.9771
8	0.7849	0.8065	0.8226	0.9613
9	0.7564	0.7752	0.7913	0.9351
10	0.7370	0.7473	0.7587	0.9242
11	0.7070	0.7229	0.7399	0.9303
12	0.6832	0.7017	0.7140	0.9204
13	0.6666	0.6828	0.6962	0.9005
14	0.6497	0.6645	0.6778	0.8768
15	0.6263	0.6431	0.6545	0.8498
$R = 500$				
5	0.8910	0.8985	0.9042	0.9912
6	0.8578	0.8699	0.8814	0.9857
7	0.8290	0.8375	0.8449	0.9799
8	0.7959	0.8053	0.8146	0.9679
9	0.7656	0.7745	0.7849	0.9492
10	0.7394	0.7483	0.7577	0.9334
11	0.7136	0.7240	0.7322	0.9360
12	0.6950	0.7025	0.7127	0.9538
13	0.6755	0.6825	0.6900	0.9209
14	0.6541	0.6635	0.6711	0.8942
15	0.6356	0.6429	0.6511	0.8739

References

[1] D.M. Blei, Probabilistic Topic Models. Communications of the ACM **55**(4), 77–84 (2012). https://doi.org/10.1145/2133806.2133826

[2] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet Allocation. Journal of Machine Learning Research **3**, 993–1022 (2003). https://doi.org/10.1162/jmlr.2003.3.4-5.993

[3] T. Hofmann, in *Proceedings of the 22nd International SIGIR-Conference* (ACM, 1999), pp. 50–57. https://doi.org/10.1145/312624.312649

[4] D.M. Blei, J.D. Lafferty, A Correlated Topic Model of Science. The Annals of Applied Statistics **1**(1), 17–35 (2007). https://doi.org/10.1214/07-AOAS114

[5] M. Rosen-Zvi, T. Griffiths, M. Steyvers, P. Smyth, in *Proceedings of the 20th UAI-Conference* (AUAI, 2004), pp. 487–494. URL https://dl.acm.org/doi/10.5555/1036843.1036902

[6] C. Wang, D.M. Blei, D. Heckerman, in *Proceedings of the 24th UAI-Conference* (AUAI, 2008), pp. 579–586

[7] M.E. Roberts, B.M. Stewart, D. Tingley, E.M. Airoldi, in *NIPS-Workshop on Topic Models: Computation, Application, and Evaluation* (2013)

[8] T.L. Griffiths, M. Steyvers, Finding scientific topics. Proceedings of the National Academy of Sciences **101**(suppl 1), 5228–5235 (2004). https://doi.org/10.1073/pnas.0307752101

[9] J. Chang, J. Boyd-Graber, S. Gerrish, C. Wang, D.M. Blei, in *NIPS: Advances in Neural Information Processing Systems*, vol. 22 (Curran Associates Inc., 2009), pp. 288–296. URL https://papers.nips.cc/paper/2009/hash/f92586a25bb3145facd64ab20fd554ff-Abstract.html

[10] A. Agrawal, W. Fu, T. Menzies, What is wrong with topic modeling? And how to fix it using search-based software engineering. Information and Software Technology **98**, 74–88 (2018). https://doi.org/10.1016/j.infsof.2018.02.005

[11] H.M. Wallach, I. Murray, R. Salakhutdinov, D. Mimno, in *Proceedings of the 26th ICML Conference* (ACM, 2009), pp. 1105–1112. https://doi.org/10.1145/1553374.1553515

[12] V.A. Nguyen, J. Boyd-Graber, P. Resnik, in *Proceedings of the 2014 EMNLP-Conference* (ACL, 2014), pp. 1752–1757. https://doi.org/10.3115/v1/D14-1182

[13] L. Koppers, J. Rieger, K. Boczek, G. von Nordheim, *tosca: Tools for Statistical Content Analysis* (2020). https://doi.org/10.5281/zenodo.3591068. R package version 0.2-0

[14] D. Mimno, H.M. Wallach, E. Talley, M. Leenders, A. McCallum, in *Proceedings of the 2011 EMNLP-Conference* (ACL, 2011), pp. 262–272

[15] K. Stevens, P. Kegelmeyer, D. Andrzejewski, D. Buttler, in *Proceedings of the 2012 Joint EMNLP/CoNLL-Conference* (ACL, 2012), pp. 952–961. URL https://aclanthology.org/D12-1087

[16] M. Röder, A. Both, A. Hinneburg, in *Proceedings of the 8th WSDM Conference* (ACM, 2015), pp. 399–408. https://doi.org/10.1145/2684822.2685324

[17] L. Xing, M.J. Paul, G. Carenini, in *Proceedings of the 2019 Joint EMNLP-IJCNLP-Conference* (ACL, 2019), pp. 3471–3477. https://doi.org/10.18653/v1/D19-1349

[18] D. Maier, A. Waldherr, P. Miltner, G. Wiedemann, A. Niekler, A. Keinert, B. Pfetsch, G. Heyer, U. Reber, T. Häussler, H. Schmid-Petri, S. Adam, Applying LDA Topic Modeling in Communication Research: Toward a Valid and Reliable Methodology. Communication Methods and Measures **12**(2-3), 93–118 (2018). https://doi.org/10.1080/19312458.2018.1430754

[19] D. Newman, E.V. Bonilla, W. Buntine, in *NIPS: Advances in Neural Information Processing Systems*, vol. 24 (Curran Associates Inc., 2011), pp. 496–504. URL https://proceedings.neurips.cc/paper/2011/hash/5ef698cd9fe650923ea331c15af3b160-Abstract.html

[20] S. Koltcov, S.I. Nikolenko, O. Koltsova, V. Filippov, S. Bodrunova, in *Internet Science*, *LNCS*, vol. 9934 (Springer, 2016), pp. 176–188. https://doi.org/10.1007/978-3-319-45982-0_16

[21] J. Rieger, ldaPrototype: A method in R to get a Prototype of multiple Latent Dirichlet Allocations. Journal of Open Source Software **5**(51), 2181 (2020). https://doi.org/10.21105/joss.02181

[22] P. Jaccard, The distribution of the flora in the alpine zone. New Phytologist **11**(2), 37–50 (1912). https://doi.org/10.1111/j.1469-8137.1912.tb05611.x

[23] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 2nd edn. Springer Series in Statistics (Springer, 2009)

[24] Y. Zhao, G. Karypis, U. Fayyad, Hierarchical clustering algorithms for document datasets. Data Mining and Knowledge Discovery **10**, 141–168 (2005). https://doi.org/10.1007/s10618-005-0361-3

[25] R Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria (2020). URL https://www.R-project.org/

[26] LexisNexis. Nexis: LexisNexis Academic & Library Solutions (2019). URL: https://www.nexis.com and https://www.lexisnexis.de/

[27] J. Rieger, G. von Nordheim, corona100d – German-language Twitter dataset of the first 100 days after Chancellor Merkel addressed the Coronavirus outbreak in TV. DoCMA Working Paper **#4** (2021). https://doi.org/10.17877/DE290R-21911

[28] I. Feinerer, K. Hornik, D. Meyer, Text Mining Infrastructure in R. Journal of Statistical Software **25**(5), 1–54 (2008). https://doi.org/10.18637/jss.v025.i05

[29] K. Benoit, K. Watanabe, H. Wang, P. Nulty, A. Obeng, S. Müller, A. Matsuo, quanteda: An R package for the quantitative analysis of textual data. Journal of Open Source Software **3**(30), 774 (2018). https://doi.org/10.21105/joss.00774

[30] B. Grün, K. Hornik, topicmodels: An R Package for Fitting Topic Models. Journal of Statistical Software **40**(13), 1–30 (2011). https://doi.org/10.18637/jss.v040.i13

[31] M.E. Roberts, B.M. Stewart, D. Tingley, stm: An R Package for Structural Topic Models. Journal of Statistical Software **91**(2), 1–40 (2019). https://doi.org/10.18637/jss.v091.i02

[32] J. Chang, *lda: Collapsed Gibbs Sampling Methods for Topic Models* (2015). URL https://CRAN.R-project.org/package=lda. R package version 1.4.2

[33] M. Lang, B. Bischl, D. Surmann, batchtools: Tools for R to work on batch systems. The Journal of Open Source Software **2**(10) (2017). https://doi.org/10.21105/joss.00135

[34] B. Bischl, M. Lang, P. Schratz, *parallelMap: Unified Interface to Parallelization Back-Ends* (2020). URL https://CRAN.R-project.org/package=parallelMap. R package version 1.5.0

[35] N. Aletras, M. Stevenson, in *Proceedings of the 14th EACL-Conference, Volume 2: Short Papers* (ACL, 2014), pp. 22–27. https://doi.org/10.3115/

v1/E14-4005

[36] D. Kim, A. Oh, in *Computational Linguistics and Intelligent Text Processing*, ed. by A. Gelbukh (Springer, 2011), pp. 163–176

[37] G. von Nordheim, J. Rieger, K. Kleinen-von Königslöw, From the fringes to the core – an analysis of right-wing populists' linking practices in seven eu parliaments and switzerland. Digital Journalism pp. 1–19 (2021). https://doi.org/10.1080/21670811.2021.1970602

[38] J. Rieger, C. Jentsch, J. Rahnenführer, in *Findings Proceedings of the 2021 EMNLP-Conference* (ACL, 2021), pp. 2337–2347. https://doi.org/10.18653/v1/2021.findings-emnlp.201

[39] D. Greene, D. O'Callaghan, P. Cunningham, in *ECML PKDD: Machine Learning and Knowledge Discovery in Databases*, *LNCS*, vol. 8724 (Springer, 2014), pp. 498–513. https://doi.org/10.1007/978-3-662-44848-9_32

[40] J. Su, D. Greene, O. Boydell, in *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)* (COLING, 2016), pp. 85–93. URL http://aclweb.org/anthology/W16-3913

[41] H.W. Kuhn, The Hungarian method for the assignment problem. Naval Research Logistics Quarterly **2**(1-2), 83–97 (1955). https://doi.org/10.1002/nav.3800020109

[42] J. Lin, Divergence measures based on the Shannon entropy. IEEE Transactions on Information Theory **37**(1), 145–151 (1991). https://doi.org/10.1109/18.61115

[43] S. Kullback, R.A. Leibler, On Information and Sufficiency. The Annals of Mathematical Statistics **22**(1), 79–86 (1951). https://doi.org/10.1214/aoms/1177729694

[44] S. Koltcov, O. Koltsova, S. Nikolenko, in *Proceedings of the 2014 WebSci Conference* (ACM, 2014), pp. 161–165. https://doi.org/10.1145/2615569.2615680

[45] M.V. Mantyla, M. Claes, U. Farooq, in *Proceedings of the 12th ACM/IEEE International ESEM-Symposium* (ACM, 2018). https://doi.org/10.1145/3239235.3267435

[46] W. Webber, A. Moffat, J. Zobel, A Similarity Measure for Indefinite Rankings. ACM Transactions on Information Systems **28**(4), 20:1–20:38 (2010). https://doi.org/10.1145/1852102.1852106

Dynamic change detection in topics based on rolling LDAs

Jonas Rieger[1], Kai-Robin Lange[1], Jonathan Flossdorf[1] and Carsten Jentsch[1]

[1]Department of Statistics, TU Dortmund University, 44221 Dortmund, Germany

Abstract

Topic modeling methods such as e.g. Latent Dirichlet Allocation (LDA) are popular techniques to analyze large text corpora. With huge amounts of textual data that are collected over time in various fields of applied research, it becomes also relevant to be able to automatically monitor the evolution of topics identified from some sort of dynamic topic modeling approach. For this purpose, we propose a dynamic change detection method that relies on a rolling version of the classical LDA that allows for coherently modeled topics over time that are able to adapt to changing vocabulary. The changes are detected by assessing the intensity of word change in the LDA's topics over time in comparison to the expected intensity of word change under stable conditions using resampling techniques. We apply our method to topics obtained by applying the RollingLDA to Covid-19 related news data from CNN and illustrate that the detected changes in these topics are well interpretable.

Keywords

change point, event, shift, narrative, story, evolution, monitoring, Latent Dirichlet Allocation

1. Introduction

While change detection is an active field in modern research, the application for text data poses even further obstacles due to its unstructured nature. And yet, an effective method for change detection would have many use cases. Particularly, when dealing with large text corpora collected over time, an online detection approach will be useful to analyze the evolution of narratives or to spot a shift in a discourse about certain topics. For this purpose, we propose an online change detection method for text data by analyzing the change of word distributions within topics of Latent Dirichlet Allocation (LDA) models. As we are dealing with time series of textual data, we make use of a rolling version of the classical LDA, called RollingLDA [1]. The method is designed to construct coherently interpretable topics modeled over time that are allowed to adapt to a changing vocabulary. The changes are detected by dynamically assessing the change intensity in word usage in the LDA's topics over time in comparison to the change intensity expected in stable periods using resampling techniques.

The main goal of change detection is to identify possible anomalies in a process. Typically,

In: R. Campos, A. Jorge, A. Jatowt, S. Bhatia, M. Litvak (eds.): Proceedings of the Text2Story'22 Workshop, Stavanger (Norway), 10-April-2022

rieger@statistik.tu-dortmund.de (J. Rieger); kalange@statistik.tu-dortmund.de (K. Lange); flossdorf@statistik.tu-dortmund.de (J. Flossdorf); jentsch@statistik.tu-dortmund.de (C. Jentsch)
0000-0002-0007-4478 (J. Rieger); 0000-0003-1172-9414 (K. Lange); 0000-0003-2153-0281 (J. Flossdorf); 0000-0001-7824-1697 (C. Jentsch)

CEUR Workshop Proceedings (CEUR-WS.org)

there are the two perspectives towards this issue: offline and online applications. Our approach is applicable for both tasks, but, for each time point, it relies exclusively on the text data that has already been observed. Hence, we focus on the usually more relevant task of online monitoring. In traditional schemes for change detection [2, 3], control charts are applied to visualize the monitoring procedure using a control statistic which is successively calculated for each time point. An alarm is triggered whenever the statistic lies outside of some control limits. In practice, there are a variety of different control charts including memory-free setups (e.g. Shewhart charts) and memory-based charts (e.g. EWMA, CUSUM), However, these traditional procedures can not be applied to textual data off the shelf because of the high dimensionality of large text corpora. In addition, an in-control state to reliably calculate the control limits is frequently not available due to the strong dynamics in text data, e.g. newspaper articles. To overcome these issues, we propose to use a control statistic based on a similarity metric that represents the resemblance of topic's word distributions over consecutive time points. Control limits are derived by a resampling procedure using word count vectors based on time-variant topics modeled by RollingLDA [1].

In a similar context, the usage of LDA was proposed for change point detection for topic distributions in texts [4], which is based on a modified version of the wild binary segmentation algorithm [5] designed for offline detection setups. There is also work considering Bayesian online monitoring [6] for textual data using a document-based model [7] and an approach based on similarity metrics, which aims to detect global events in topics in offline settings [8]. There is also work which analyzes the transitions of narratives between topics [9]. In contrast, the rolling window approach of RollingLDA constructs coherently interpretable topics modeled over time and allows the resulting dynamic change detection method to become applicable in online settings. Compared to the mentioned related methods, our method is designed to detect changes in word distributions of topics over time rather than global changes in topic distributions (of sets) of documents [e.g. 4, 7, 8] or sentiments in topics [e.g. 10] or (in contrast) changes in topic distributions of words [e.g. 11]. This results in a more refined monitoring procedure that allows for the detection of narrative shifts that are changing the word usage within a certain topic instead of measuring the frequency of a topic over time within the whole corpus. Building on this, we aim that our proposed method can provide groundwork for the extraction and temporal localization of narratives in texts.

2. Methodological framework

For the proposed change detection algorithm, we make use of the existing method of a rolling version of the classical LDA (RollingLDA) to construct coherent topics over time and measure similarities of topics for consecutive time points using the well-established cosine similarity.

2.1. Latent Dirichlet Allocation

The classical LDA [12] models distributions of K latent topics for each text. Let $W_n^{(m)}$ be a single word token at position $n = 1, \dots N^{(m)}$ in text $m = 1, \dots, M$ of a corpus of M texts. Then, a single

text is given by

$$\boldsymbol{D}^{(m)} = \left(W_1^{(m)}, \dots, W_{N^{(m)}}^{(m)}\right), \quad W_n^{(m)} \in \boldsymbol{W} = \{W_1, \dots, W_V\}, V = |\boldsymbol{W}|$$

and the corresponding topic assignments for each text are given by

$$\boldsymbol{T}^{(m)} = \left(T_1^{(m)}, \dots, T_{N^{(m)}}^{(m)}\right), \quad T_n^{(m)} \in \boldsymbol{T} = \{T_1, \dots, T_K\}.$$

From this, let $n_k^{(mv)}, k = 1, \dots, K, v = 1, \dots, V$ denote the number of assignments of word v in text m to topic k. Then, we define the cumulative count of word v in topic k over all texts by $n_k^{(\bullet v)}$ and denote the total count of assignments to topic k by $n_k^{(\bullet\bullet)}$. Using these definitions, the underlying probability model [13] can be written as

$$W_n^{(m)} \mid T_n^{(m)}, \phi_k \sim \text{Discr}(\phi_k), \quad \phi_k \sim \text{Dir}(\eta), \quad T_n^{(m)} \mid \theta_m \sim \text{Discr}(\theta_m), \quad \theta_m \sim \text{Dir}(\alpha).$$

For a given parameter set $\{K, \alpha, \eta\}$, with the Dirichlet priors α and η defining the type of mixture of topics in every text and the type of mixture of words in every topic, LDA assigns one of the K topics to each token. A word distribution estimator per topic for $\phi_k = (\phi_{k,1}, \dots, \phi_{k,V})^T \in (0,1)^V$ can be derived through the collapsed Gibbs sampler procedure [13] by

$$\hat{\phi}_{k,v} = \frac{n_k^{(\bullet v)} + \eta}{n_k^{(\bullet\bullet)} + V\eta}. \tag{1}$$

2.2. RollingLDA

RollingLDA [1] is a rolling version of classical LDA. New texts are modeled based on existing topics of the previous model. Thereby, not the whole knowledge of the entire past of the model is used, but only the information of the topics from more recent texts based on a user-chosen memory parameter. For each time point, based on the topic assignments within this memory period, the topics are initialized and modeled forward. This form of modeling preserves the topic structure of the model so that topics remain coherently interpretable over time. At the same time, constraining the knowledge of the model to the user-chosen memory period allows for changes in topics based on new vocabulary or word choices. There are other dynamic variants of the LDA approach [14, 15, 16, 17, 18] deliberately designed to model gradual changes, and therefore not as well suited to detect abrupt changes. We use the update algorithm RollingLDA to make our proposed change detection method applicable in an online manner. Thereby, a text is assigned to a time point on the basis of its publication date. The step size of the and is chosen on a weekly basis in the present case as this seems natural for journalistic texts.

2.3. Similarity

Our change detection algorithm builds on a similarity measure for word count vectors. Following up on the notation from Section 2.1 the word count vector for topic $k \in \{1, \dots, K\}$ at one time point $t \in \{0, \dots, T\}$ is given by

$$\boldsymbol{n}_{k|t} = \left(n_{k|t}^{(\bullet 1)}, \dots, n_{k|t}^{(\bullet V)}\right)^T \in \mathbb{N}_0^V = \{0, 1, 2, \dots\}^V.$$

Then, monitoring the similarity of topics over time for (consecutive) time points t_1 and t_2 is done using the cosine similarity

$$\cos\left(\boldsymbol{n}_{k|t_1}, \boldsymbol{n}_{k|t_2}\right) = \frac{\sum_v n_{k|t_1}^{(\bullet v)} n_{k|t_2}^{(\bullet v)}}{\sqrt{\sum_v \left(n_{k|t_1}^{(\bullet v)}\right)^2} \sqrt{\sum_v \left(n_{k|t_2}^{(\bullet v)}\right)^2}}. \tag{2}$$

The choice of cosine similarity is common in the context of change point detection for text data [e.g. 8, 19]. Compared to other similarity measures such as the Jaccard coefficient, Jensen-Shannon Divergence, χ^2-, Hellinger and Manhattan Distance, the cosine similarity fulfills some typical conditions required for monitoring a similarity measure [1].

3. Change detection

In combination with the existing method RollingLDA and cosine similarity, our contributed method for change detection relies on classical resampling approaches to identify changes within topics. We estimate the realized change in a topic based on the similarity between the current and previous count vectors of word assignments and compare the resulting similarity score to resampling-based similarity scores which are generated under stable conditions, such that no extraordinary changes occurred in the topic.

3.1. Set of changes

Suppose we consider K topics over T time points to be monitored. If the actual observed similarity of the word vector of some topic $k \in \{1, \dots, K\}$ at some time $t \in \{0, 1, \dots, T\}$ given by $\boldsymbol{n}_{k|t}$, compared to the frequency vector of the topic over a predefined reference time period $t - z_k^t, \dots, t-1$, given by

$$\boldsymbol{n}_{k|(t-z_k^t):(t-1)} = \sum_{z=1}^{z_k^t} \boldsymbol{n}_{k|t-z}, \tag{3}$$

is smaller than a threshold q_k^t which is calibrated based on similarities under stable conditions (see Section 3.2), then we identify a change within topic k at time t. The set of identified changes in topic k up to time point t can then be defined as

$$C_k^t = \left\{u \mid 0 < u \leq t \leq T \,:\, \cos\left(\boldsymbol{n}_{k|u}, \boldsymbol{n}_{k|(u-z_k^u):(u-1)}\right) < q_k^t\right\} \cup 0, \tag{4}$$

where the time point $t = 0$ is always included for technical reasons, to compute the current run length without a change $z_k^t = \min\left\{z_{\max}, t - \max C_k^{t-1}\right\}$. Thus, the reference period spans the last $z_{\max}$ time points if no change was detected during that time, and spans the time that has passed since the last change, otherwise. The parameter $z_{\max}$ is to be chosen by the user and is intended to smooth the similarities to prevent from detecting false positives.

3.2. Dynamic thresholds

For the calculation of the threshold q_k^t, the estimated word distribution of a topic k at some time point t, as well as over the corresponding reference period $t - z_k^t, \dots, t-1$ are needed. For this, let $\hat{\phi}_k^t$ and $\hat{\phi}_k^{(t-z_k^t):(t-1)}$ be defined by

$$\hat{\phi}_{k,v}^t = \frac{n_{k|t}^{(\bullet v)} + \eta}{n_{k|t}^{(\bullet\bullet)} + V\eta} \quad \text{and} \quad \hat{\phi}_{k,v}^{(t-z_k^t):(t-1)} = \frac{n_{k|(t-z_k^t):(t-1)}^{(\bullet v)} + \eta}{n_{k|(t-z_k^t):(t-1)}^{(\bullet\bullet)} + V\eta} \tag{5}$$

analogously to Equation (1).

The application of the change point detection algorithm is designed for text data, more precisely for empirical word distributions of K topics modeled by LDA in a given text corpus. Since word choice - especially in journalistic texts - varies considerably over time, a situation in which there is no change in the word distribution within topics across consecutive time points does not reflect the expected situation. Rather, it is to be expected that topics change gradually on an ongoing basis. Accordingly, our method aims to identify not the numerous customary changes in the topics, but the unexpectedly large ones. To do so, we define an expected word distribution $\tilde{\phi}_k^{(t)}$ for time point t under stable conditions that include the customary changes as a convex combination of the two estimators of the word distribution of topic k, one for the reference time period $t - z_k^t, \dots, t-1$ and one for the current time point t. Using the mixture parameter $p \in [0,1]$, which can be tuned based on how substantial the detected changes should be, the intensity of the expected change is considered in the determination of this estimator by

$$\tilde{\phi}_k^{(t)} = (1-p)\,\hat{\phi}_{k,v}^{(t-z_k^t):(t-1)} + p\,\hat{\phi}_{k,v}^{(t)}. \tag{6}$$

Our method uses the estimator $\tilde{\phi}_k^{(t)}$ to simulate R expected word count vectors $\tilde{\boldsymbol{n}}_{k|t}^r, r = 1, \dots, R$ based on a parametric bootstrap approach. In this process, each word is drawn according to its estimated probability of occurrence regarding $\tilde{\phi}_k^{(t)}$ and each sample r consists of $n_{k|t}^{(\bullet\bullet)}$ draws, the number of words assigned to topic k at time point t. Then, we calculate the cosine similarity

$$\cos\left(\tilde{\boldsymbol{n}}_{k|t}^r, \boldsymbol{n}_{k|(t-z_k^t):(t-1)}\right) \tag{7}$$

for each of the $r = 1, \dots, R$ bootstrap samples and set the threshold q_k^t equal to the 0.01 quantile of these simulated similarity values generated under stable conditions. Combinations of topics and time points for which the observed similarity is smaller than the corresponding quantile are classified as change points according to Equation (4).

4. Analysis

For conducting the real data analysis, the data set under study was created with Python, whereas the preprocessing, the modeling, all postprocessing steps and analyses are performed using R. The scripts for all analysis steps can be found in the associated GitHub repository github.com/JonasRieger/topicalchanges.

4.1. Data and study design

To assess the quality of our change point algorithm, we use the TLS-Covid19 data set [20]. It is generated using Covid-19 related liveblog articles of CNN, collected from January 22nd 2020 up until December 12th 2021. Each liveblog is interpreted to belong to a topic and comprises texts and key moments. The texts form a time line containing events, which are summarized by its key moments. The resulting corpus consists of 27,432 texts and 1,462 key moments. Although the data set contains multiple key moments per day on average, we do not consider all them a change point as our aim is to detect larger changes based on aggregated weekly texts. However, these key moments serve well as indicators, which enable us to check whether the detected changes are actually related to real events or if they are false positives.

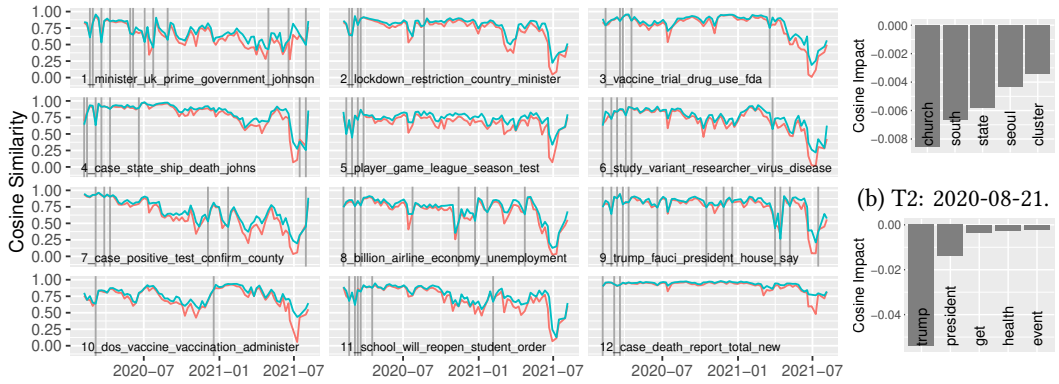
We use common NLP preprocessing steps for the texts, i.e. characters are formatted to lowercase, numbers and punctuation are removed. Moreover, a trusted stopword list is applied to remove words that do not help in classifying texts in topics, we use a lemmatization dictionary (github.com/michmech/lemmatization-lists) and neglect words with less than two characters.

We model the CNN data set using RollingLDA on a weekly basis, starting on Saturday of each week, and we consider the previous week as initialization for the model's topics. The first 10 days of modeling, Wednesday, January 22nd 2020 until Friday, January 31st 2020, serve as the initial chunk corresponding to $t = 0$. During this period, 605 texts were published. In the data set, there are weeks that do not contain any texts. In this case, the corresponding time point is omitted. Then, to model the texts of the following chunk, at least the last 10 texts are used, as well as all other texts published on the same date as the oldest of these 10 texts. As parameters, we assume $K = 12$ topics, define the reference period of the topics to the last $z_{\max} = 4$ weeks, and choose $p = 0.85$, since these values are accountable by plausibility and seem to yield reasonable results. For other parameter choices, i.e. $K = 8, \dots, 20$, $z_{\max} = 1, \dots, 20$, $p = 0.5, \dots 0.8, 0.81, \dots, 0.90$, results can be found in our associated repository.

4.2. Findings

The results of our chosen model are displayed in Figure 1. Fig. 1a shows the detected changes by vertical gray lines, which are the weeks in which the observed similarity (blue curve) is lower than the expected one (red curve). Furthermore, for two changes we show which words are mainly causing the detection of the change. The score of a word in a topic at a given time point is calculated by the topic's similarity without considering this word and subtracting it from the actual realized similarity. These leave-one-out cosine impact scores for the words with the five most negative scores are shown in Fig. 1b and 1c. In general, most of the changes we detect occur within the first four months of 2020. This is because the wording was constantly changing, as the Covid-19 epidemic turned into a pandemic over the course of these months. New people and organizations were associated with Covid-19, which is why we detect a bunch of consecutive changes in every topic. As the pandemic reached out into further countries, the detected changes became less frequent for most topics. In the following we share our interpretation of some exemplary detected changes.

The third topic, containing information about vaccination and testing procedures, shows a change in the week starting on the 13th of March 2021. In this week, the AstraZeneca



(a) Similarities (blue), thresholds (red) and detected changes (vertical lines, gray).

(b) T2: 2020-08-21.

(c) T9: 2020-10-16.

Figure 1: Similarity values, thresholds, and detected changes over the observation period for all $K = 12$ topics, as well as the five most influential words for two selected change points in topics 2 and 9.

vaccination process in several EU-states was stopped due the risk of causing blood clots.[1] The sixth topic, a topic about medical studies and research, shows a change in the following week, in which AstraZeneca presented a study about the effectiveness of its vaccine.[2] Another interesting detection is the change in the vaccination-related topic 10 in December 2020, just as the vaccination process started in the US.[3]

Political changes are also detected in several topics, such as the start of Joe Biden's presidential era in late January 2021 in topic 11, the return of Donald Trump to office after his Covid-19 infection in October 2020[4] in topic 9 (cf. Fig. 1c) or the discussion about the origin of the virus after a WHO report in late March 2021 in topic 9.[5] A Covid-19 outbreak in the South Korean Sarang-jeil church in August 2020[6] is detected in topic 2 (cf. Fig. 1b).

While these topics detect changes across the entire time span, the twelfth topic, representing the report of the current number of Covid cases, does not detect a single change after March

[1]CNN online, 2021-03-15 3:03 p.m. ET, "Spain joins Germany, France and Italy in halting AstraZeneca Covid-19 vaccinations", https://edition.cnn.com/world/live-news/coronavirus-pandemic-vaccine-updates-03-15-21/h_d938057f2ef588f74565bdbb01f12387, visited on 2022-01-20.

[2]CNN online, 2021-03-25 2:48 a.m. ET, "New AstraZeneca report says vaccine was 76% effective in preventing Covid-19 symptoms", https://edition.cnn.com/world/live-news/coronavirus-pandemic-vaccine-updates-03-25-21/h_9f01e2e53b62873f1c742254d27fbf5f, visited on 2022-01-20.

[3]CNN online, 2020-12-14 10:08 p.m. ET, "The first doses of FDA-authorized Covid-19 vaccine were administered in the US. Here's what we know", https://edition.cnn.com/world/live-news/coronavirus-pandemic-vaccine-updates-12-15-20/h_32be1a72dc05f874eda167c95c8f1bba, visited on 2022-01-20.

[4]CNN online, 2020-10-12 12:01 a.m. ET, "Trump says he tested 'totally negative' for Covid-19", https://edition.cnn.com/world/live-news/coronavirus-pandemic-10-12-20-intl/h_7570d53b184a5b1d6ec97ce67330e4c9, visited on 2022-01-20.

[5]CNN online, 2021-03-29 11:22 a.m. ET, "Upcoming WHO report will deem Covid-19 lab leak extremely unlikely, source says", https://www.cnn.com/world/live-news/coronavirus-pandemic-vaccine-updates-03-29-21/h_1f239fee1b0584ca9a5b6085357ac907, visited on 2022-01-20.

[6]CNN online, 2020-08-20 12:55 a.m. ET, "South Korea's latest church-linked coronavirus outbreak is turning into a battle over religious freedom", https://edition.cnn.com/world/live-news/coronavirus-pandemic-08-20-20-intl/h_288a15acd1b29e732c4e10693641088a, visited on 2022-01-20.

2020. This is most likely because, after the pandemic had reached the US and Europe in early 2020, the number of cases was consistently reported and the interpretations and implication of those case numbers are detected as changes in other topics. Even in the last months of the data set, in which the number of texts decreased and the results thus show a lower similarity, the twelfth topic retained a rather high similarity of above 0.75.

5. Discussion

In this paper, we presented a novel change detection method for text data. To construct coherently interpretably topics, we used RollingLDA to model a time series on textual data and compared the model's word distribution vectors with those of texts resampled under stable conditions. We applied our model on the TLS-Covid19 data set consisting of Covid-19 related news articles from CNN between January 2020 and December 2021.

Our method detects several meaningful changes in the evolving news coverage during the pandemic, including e.g. the start of vaccinations and several controversies over the course of the vaccination campaign as well as political changes such as the start of Joe Biden's presidential era. Out of 78 detected changes, we were instantly able to judge 55 (71%) as plausible ones based on manual labeling using the leave-one-out cosine impacts (cf. Fig. 1b, 1c and repository). The share increases to 78% if we exclude the turbulent initial phase of the Covid-19 pandemic and only consider changes since April 2020. While we cannot tell how many changes were missed out that could be considered as important as the ones mentioned above, our model contains a mixture parameter to calibrate the detection for general change of topics within a usual news week. If more, but less substantial or less, but more substantial changes are to be detected, this parameter p can be tuned accordingly. In combination with the maximum length of the reference period $z_{\max}$, the set $\{p, z_{\max}\}$ forms the model's hyperparameters to be optimized.

Acknowledgments

The present study is part of a project of the Dortmund Center for data-based Media Analysis (DoCMA) at TU Dortmund University. The work was supported by the Mercator Research Center Ruhr (MERCUR) with project number PR-2019-0019. In addition, the authors gratefully acknowledge the computing time provided on the Linux HPC cluster at TU Dortmund University (LiDO3), partially funded in the course of the Large-Scale Equipment Initiative by the German Research Foundation (DFG) as project 271512359.

References

[1] J. Rieger, C. Jentsch, J. Rahnenführer, RollingLDA: An update algorithm of Latent Dirichlet Allocation to construct consistent time series from textual data, in: Findings Proceedings of the 2021 EMNLP-Conference, ACL, 2021, pp. 2337–2347. doi:10.18653/v1/2021.findings-emnlp.201.

[2] D. C. Montgomery, Introduction to statistical quality control, John Wiley & Sons, 2020.

[3] J. S. Oakland, Statistical process control, Routledge, 2007.

[4] A. Bose, S. S. Mukherjee, Changepoint analysis of topic proportions in temporal text data, 2021. arXiv:2112.00827.

[5] P. Fryzlewicz, Wild binary segmentation for multiple change-point detection, The Annals of Statistics 42 (2014) 2243–2281. doi:10.1214/14-AOS1245.

[6] R. P. Adams, D. J. MacKay, Bayesian online changepoint detection, 2007. arXiv:0710.3742.

[7] T. Kim, J. Choi, Reading documents for bayesian online change point detection, in: Proceedings of the 2015 EMNLP-Conference, ACL, 2015, pp. 1610–1619. doi:10.18653/v1/D15-1184.

[8] N. Keane, C. Yee, L. Zhou, Using topic modeling and similarity thresholds to detect events, in: Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation, ACL, 2015, pp. 34–42. doi:10.3115/v1/W15-0805.

[9] Q. Mei, C. Zhai, Discovering evolutionary theme patterns from text: An exploration of temporal text mining, in: Proceedings of the 11th SIGKDD-Conference, ACM, 2005, pp. 198–207. doi:10.1145/1081870.1081895.

[10] Q. Liang, K. Wang, Monitoring of user-generated reviews via a sequential reverse joint sentiment-topic model, Quality and Reliability Engineering International 35 (2019) 1180–1199. doi:10.1002/qre.2452.

[11] L. Frermann, M. Lapata, A Bayesian model of diachronic meaning change, Transactions of the Association of Computational Linguistics 4 (2016) 31–45. doi:10.1162/tacl_a_00081.

[12] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent Dirichlet Allocation, Journal of Machine Learning Research 3 (2003) 993–1022. doi:10.1162/jmlr.2003.3.4-5.993.

[13] T. L. Griffiths, M. Steyvers, Finding scientific topics, Proceedings of the National Academy of Sciences 101 (2004) 5228–5235. doi:10.1073/pnas.0307752101.

[14] X. Song, C.-Y. Lin, B. L. Tseng, M.-T. Sun, Modeling and predicting personal information dissemination behavior, in: Proceedings of the 11th SIGKDD-Conference, ACM, 2005, pp. 479–488. doi:10.1145/1081870.1081925.

[15] D. M. Blei, T. L. Griffiths, M. I. Jordan, J. B. Tenenbaum, Hierarchical topic models and the nested chinese restaurant process, in: Advances in Neural Information Processing Systems, volume 16, MIT Press, 2003, pp. 17–24. URL: https://proceedings.neurips.cc/paper/2003/hash/7b41bfa5085806dfa24b8c9de0ce567f-Abstract.html.

[16] X. Wang, A. McCallum, Topics over time: A non-markov continuous-time model of topical trends, in: Proceedings of the 12th SIGKDD-Conference, ACM, 2006, pp. 424–433. doi:10.1145/1150402.1150450.

[17] D. M. Blei, J. D. Lafferty, Dynamic topic models, in: Proceedings of the 23rd ICML-Conference, ACM, 2006, pp. 113–120. doi:10.1145/1143844.1143859.

[18] C. Wang, D. M. Blei, D. Heckerman, Continuous time dynamic topic models, in: Proceedings of the 24th UAI-Conference, AUAI Press, 2008, pp. 579–586. URL: https://dl.acm.org/doi/10.5555/3023476.3023545.

[19] Y. Wang, C. Goutte, Real-time change point detection using on-line topic models, in: Proceedings of the 27th ACL-Conference, ACL, 2018, pp. 2505–2515. URL: https://www.aclweb.org/anthology/C18-1212.

[20] A. Pasquali, R. Campos, A. Ribeiro, B. Santana, A. Jorge, A. Jatowt, TLS-Covid19: A new annotated corpus for timeline summarization, in: Advances in Information Retrieval, ECIR 2021, volume 12656 of *LNCS*, 2021, pp. 497–512. doi:10.1007/978-3-030-72113-8_33.

Improving Latent Dirichlet Allocation: On Reliability of the Novel Method LDAPrototype

Jonas Rieger[0000−0002−0007−4478], Jörg Rahnenführer[0000−0002−8947−440X], and Carsten Jentsch[0000−0001−7824−1697]

Department of Statistics, TU Dortmund University, 44221 Dortmund, Germany
{rieger,rahnenfuehrer,jentsch}@statistik.tu-dortmund.de

Abstract. A large number of applications in text data analysis use the Latent Dirichlet Allocation (LDA) as one of the most popular methods in topic modeling. Although the instability of the LDA is mentioned sometimes, it is usually not considered systematically. Instead, an LDA is often selected from a small set of LDAs using heuristic means or human codings. Then, conclusions are often drawn based on the to some extent arbitrarily selected model. We present the novel method LDAPrototype, which takes the instability of the LDA into account, and show that by systematically selecting an LDA it improves the reliability of the conclusions drawn from the result and thus provides better reproducibility. The improvement coming from this selection criterion is unveiled by applying the proposed methods to an example corpus consisting of texts published in a German quality newspaper over one month.

Keywords: Topic Model · Machine Learning · Similarity · Stability · Stochastic

1 Introduction

Due to the growing number and especially the increasing amount of unstructured data, it is of great interest to be able to analyze them. Text data is an example for unstructured data and at the same time it covers a large part of them. It is organized in so-called corpora, which are given by collections of texts.

For the analysis of such text data topic models in general and the Latent Dirichlet Allocation in particular is often used. This method has the weakness that it is unstable, i.e. it gives different results for repeated runs. There are various approaches to reduce this instability. In the following, we present a new method LDAPrototype that improves the reliability of the results by choosing a center LDA. We will demonstrate this improvement of the LDA applying the method to a corpus consisting of all articles published in the German quality newspaper Süddeutsche Zeitung in April 2019.

1.1 Related Work

The Latent Dirichlet Allocation [3] is very popular in text data analysis. Numerous extensions to Latent Dirichlet Allocation have been proposed, each customized for certain applications, as the Author-Topic Model [18], Correlated

Topics Model [2] or the more generalized Structural Topic Model [17]. We focus on LDA as one of the most commonly used topic models and propose a methodology to increase reliability of findings drawn from the results of LDA.

Reassigning words to topics in the LDA is based on conditional distributions, thus it is stochastic. This is rarely discussed in applications [1]. However, several approaches exist to encounter this problem based on a certain selection criterion. One of these selection criteria is perplexity [3], a performance measure for probabilistic models to estimate how well new data fit into the model [18]. As an extension, Nguyen et al. [13] proposed to average different iterations of the Gibbs sampling procedure to achieve an increase of perplexity. In general, it was shown that optimizing likelihood-based measures like perplexity does not select the model that fits the data best regarding human judgements. In fact, these measures are negatively correlated with human judgements on topic quality [5]. A better approach should be to optimize semantic coherence of topics as Chang et al. [5] proposed. They provide a validation technique called Word or Topic Intrusion which depends on a coding process by humans. Measures without human interaction, but almost automated, and also aiming to optimize semantic coherence can be transferred from the Topic Coherence [12]. Unfortunately, there is no validated procedure to get a selection criterion for LDA models from this topic's "quality" measure. Instead, another option to overcome the weakness of instability of LDA is to start the first iteration of the Gibbs sampler with reasonably initialized topic assignments [11] of every token in all texts. One possibility is to use co-occurences of words. The initialization technique comes with the drawback of restricting the model to a subset of possible results.

1.2 Contribution

In this paper, we propose an improvement of the Latent Dirichlet Allocation through a selection criterion of multiple LDA runs. The improvement is made by increasing the reliability of results taken from LDA. This particular increase is obtained by selecting the model that represents the center of the set of LDAs best. The method is called LDAPrototype [16] and is explained in Section 3. We show that it generates reliable results in the sense that repetitions lie in a rather small sphere around the overall centered LDA, when applying the proposed methods to an example corpus of articles from the Süddeutsche Zeitung.

2 Latent Dirichlet Allocation

The method we propose is based on the LDA [3] estimated by a Collapsed Gibbs sampler [6], which is a probabilistic topic model that is widely used in text data analysis. The LDA assumes that there is a topic distribution for every text, and it models them by assigning one topic from the set of topics $T = \{T_1, ..., T_K\}$ to every token in a text, where $K \in \mathbb{N}$ denotes the user-defined number of modeled topics. We denote a text (or document) of a corpus consisting of M texts by

$$\boldsymbol{D}^{(m)} = \left(W_1^{(m)}, ..., W_{N^{(m)}}^{(m)}\right), \; m = 1, ..., M, \quad W_n^{(m)} \in \boldsymbol{W}, \; n = 1, ..., N^{(m)}.$$

We refer to the size of text m as $N^{(m)}$; $\boldsymbol{W} = \{W_1, ..., W_V\}$ is the set of words and $V \in \mathbb{N}$ denotes the vocabulary size. Then, analogously the topic assignments of every text m are given by

$$\boldsymbol{T}^{(m)} = \left(T_1^{(m)}, ..., T_{N^{(m)}}^{(m)}\right), \ m = 1, ..., M, \qquad T_n^{(m)} \in T, \ n = 1, ..., N^{(m)}.$$

Each topic assignment $T_n^{(m)}$ corresponds to the token $W_n^{(m)}$ in text m. When $n_k^{(mv)}, k = 1, ..., K, v = 1, ..., V$ describes the number of assignments of word v in text m to topic k, we can define the cumulative count of word v in topic k over all documents by $n_k^{(\bullet v)}$. Then, let $\boldsymbol{w}_k = (n_k^{(\bullet 1)}, ..., n_k^{(\bullet V)})^T$ denote the vectors of word counts for the $k = 1, ..., K$ topics. Using these definitions, the underlying probability model of LDA [6] can be written as

$$\begin{aligned} W_n^{(m)} \mid T_n^{(m)}, \boldsymbol{\phi}_k &\sim \text{ Discrete}(\boldsymbol{\phi}_k), & \boldsymbol{\phi}_k &\sim \text{ Dirichlet}(\eta), \\ T_n^{(m)} \mid \boldsymbol{\theta}_m &\sim \text{ Discrete}(\boldsymbol{\theta}_m), & \boldsymbol{\theta}_m &\sim \text{ Dirichlet}(\alpha), \end{aligned}$$

where α and η are Dirichlet distribution hyperparameters and must be set by the user. Although the LDA permits α and η to be vector valued [3], they are usually chosen symmetric because typically the user has no a-priori information about the topic distributions $\boldsymbol{\theta}$ and word distributions $\boldsymbol{\phi}$. Increasing η leads to a loss of homogenity of the mixture of words per topic. In contrast, a decrease leads to a raise of homogenity, identified by less but more dominant words per topic. In the same manner α controls the mixture of topics in texts.

3 LDAPrototype

The Gibbs sampler in the modeling procedure of the LDA is sensitive to the random initialization of topic assignments as mentioned in Section 1.1. We present a method that reduces the stochastic component of the LDA. This adaption of the LDA named LDAPrototype [16] increases the reliability of conclusions drawn from the resulting prototype model, which is obtained by selecting the model that seems to be the most central of (usually around) 100 independently modeled LDA runs. The procedure can be compared to the calculation of the median in the univariate case.

The method makes use of topic similarities measured by the modified Jaccard coefficient for the corresponding topics to the word count vectors $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$

$$J_m(\boldsymbol{w}_i, \boldsymbol{w}_j) = \frac{\sum_{v=1}^{V} \mathbb{1}_{\left\{n_i^{(\bullet v)} > c_i \ \wedge \ n_j^{(\bullet v)} > c_j\right\}}}{\sum_{v=1}^{V} \mathbb{1}_{\left\{n_i^{(\bullet v)} > c_i \ \vee \ n_j^{(\bullet v)} > c_j\right\}}},$$

where $\boldsymbol{c}$ is a vector of lower bounds. Words are assumed to be relevant for a topic if the count of the word passes this bound. The threshold $\boldsymbol{c}$ marks the

modification to the traditional Jaccard coefficient [8] and can be chosen in an absolute or relative manner or as a combination of both.

The main part of LDAPrototype is to cluster two independent LDA replications using Complete Linkage [7] based on the underlying topic similarities of those two LDA runs. Let G be a pruned cluster result composed by single groups g consisting of topics and let $g_{|1}$ and $g_{|2}$ denote groups of g restricted to topics of the corresponding LDA run. Then, the method aims to create a pruning state where $g_{|1}$ and $g_{|2}$ are each build by only one topic for all $g \in G$. This is achieved by maximizing the measure for LDA similarity named S-CLOP (**S**imilarity of Multiple Sets by **C**lustering with **Lo**cal **P**runing) [16]:

$$\text{S-CLOP}(G) = 1 - \frac{1}{2K} \sum_{g \in G} |g| \left(||g_{|1}| - 1| + ||g_{|2}| - 1| \right) \in [0, 1].$$

We denote the best pruning state by $G^* = \arg\max\{\text{S-CLOP}(G)\}$ for all possible states G and determine similarity of two LDA runs by S-CLOP(G^*). The prototype model of a set of LDAs then is selected by maximizing the mean pairwise similarity of one model to all other models.

The methods are implemented in the R [14] package ldaPrototype [15]. The user can specify the number of models, various options for $\boldsymbol{c}$ including a minimal number of relevant words per topic as well as the necessary hyperparameters for the basic LDA α, η, K and the number of iterations the Gibbs sampler should run. The package is linked to the packages lda [4] and tosca [10].

4 Analysis

We show that the novel method LDAPrototype improves the Latent Dirichlet Allocation in the sense of reliability. To prove that, the following study design is applied to an example corpus from the German quality newspaper Süddeutsche Zeitung (SZ). The corpus consists of all 3 718 articles published in the SZ in April 2019. It is preprocessed using common steps for cleaning text data including duplicate removal leading to 3 468 articles. Moreover, punctuation, numbers and German stopwords are removed. In addition, all words that occur ten times or less are deleted. This results in $M = 3\,461$ non-empty texts and a vocabulary size of $V = 11\,484$. The preprocessing was done using the R package tosca [10].

4.1 Study Design

The study is as follows: First of all, a large number N of LDAs is fitted. This set represents the basic population of all possible LDAs in the study. Then we repeat P times the random selection of R LDAs and calculate their LDAPrototype. This means, finally P prototypes are selected, each based on R basic LDAs, where each LDA is randomly drawn from a set of N LDAs. Then, a single prototype is determined based on a comparison of the P prototypes. This particular prototype forms the assumed true center LDA. In addition, we establish a ranking of all

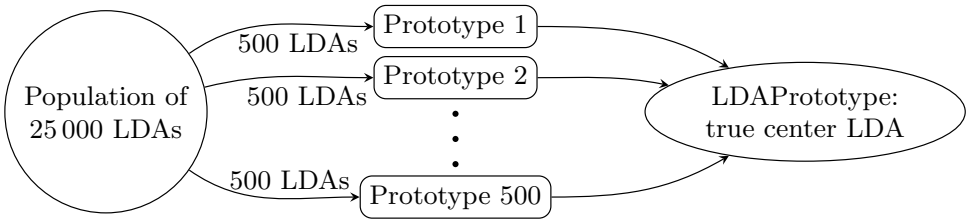


Fig. 1: Schematic representation of the study design for $N = 25\,000$ LDAs in the base population and $P = 500$ selected prototypes, each based on $R = 500$ sampled LDAs from the base population.

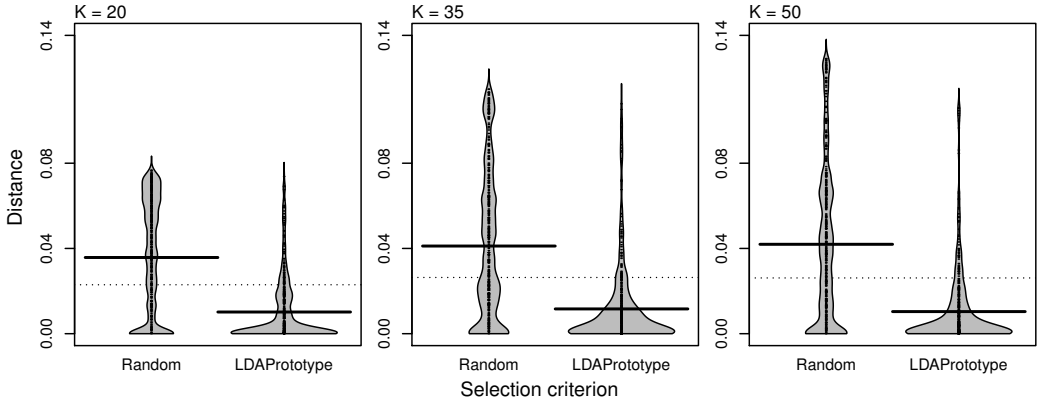
other prototypes. The order is determined by sequentially selecting the next best prototype which realizes the maximum of the mean S-CLOP values by adding the corresponding prototype and simultaneously considering all higher ranked LDAPrototypes.

For the application we choose three different parameter combinations for the basic LDA. In fact, we want to model the corpus of the SZ with $K = 20, 35, 50$ topics. We choose accordingly $\alpha = \eta = 1/K$ and let the Gibbs sampler iterate 200 times. We choose the size of the population as $N = 25\,000$, so that we initially calculate a total of 75 000 LDAs, which is computationally intensive but bearable. We use the R package ldaPrototype [15] to compute the models on batch systems. We set the parameters of the study to a sufficiently high and at the same time calculable value of $P = R = 500$. That is, we get 500 PrototypeLDAs, each based on 500 basic LDAs, that are sampled without replacement from the set of 25 000 basic LDAs. The sampling procedure is carried out without replacement in order to protect against falsification by multiple selection of one specific LDA. Figure 1 represents this particular study design schematically.

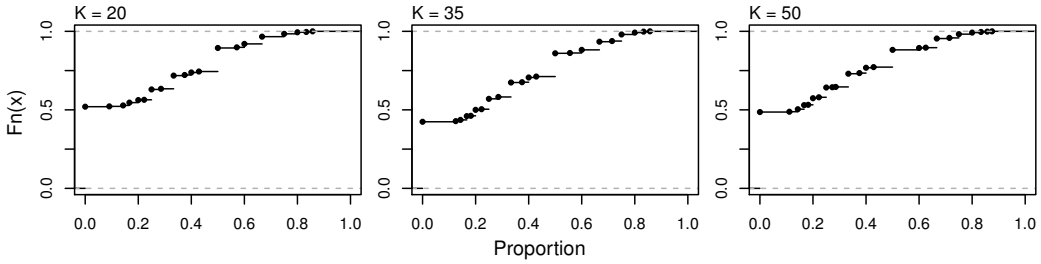
Then, we inspect the selection of the P prototypes. On the one hand, we quantify the goodness of selection by determining how many LDAs, that were available in the corresponding run, are ranked before the corresponding LDAPrototype. On the other hand, the analysis of the distance to the best available LDA run in the given prototype run provides a better assessment of the reliability of the method. We compare the observed values with randomized choices of the prototype. This leads to statements of the form that the presented method LDAPrototype selects its prototypes only from a sufficiently small environment around the true center LDA, especially in comparison to random selected LDAs.

4.2 Results

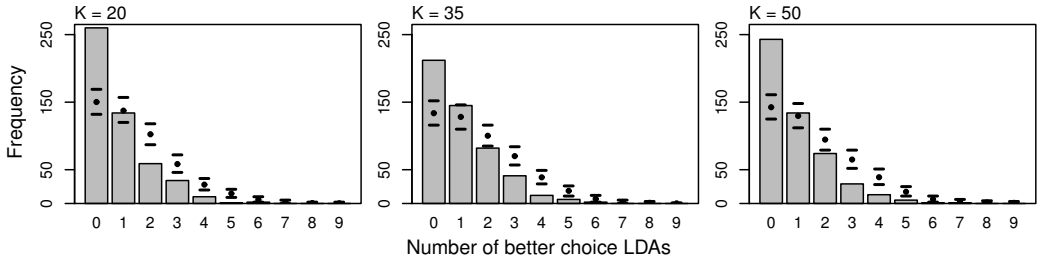
For the analysis we first determine the true center LDA and a ranking for all 500 prototypes as described in Section 4.1 for each $K = 20, 35, 50$. The corresponding mean S-CLOP value at the time of addition is assigned to each prototype in the ranking as a measure of proximity to the true center LDA. To visualize the rankings, we use so-called beanplots [9] as a more accurate form of boxplots, as well as empirical cumulative distribution functions (ECDF) and bar charts.



(a) Distance of each of the LDAPrototypes to the LDA that would have been the best choice in the corresponding prototype run regarding closeness to the center LDA.



(b) Empirical cumulative distribution function of the proportion of how many LDAs are closer to the center LDA than the selected LDAPrototype.



(c) Number of LDAs that are closer to the center LDA than the selected LDAPrototype.

Fig. 2: Analysis of the improvement of reliability by using the LDAPrototype for $K = 20, 35, 50$ modeled topics. Every single value corresponds to one of the $P = 500$ prototype runs resulting in the corresponding LDAPrototype.

For $K = 20, 35, 50$ each of the 25 000 LDAs is included at least once in the 500 times 500 selected LDAs. Nevertheless, only 169, 187 and 186 different LDAs are chosen as prototypes. The LDAPrototype method thus differs significantly from a random selection, whose associated simulated 95% confidence interval suggests between 490 and 499 different prototypes.

Figure 2 summarizes the analysis of the increase of reliability for 20, 35 and 50 topics, respectively. The beanplots in Figure 2a indicate the distance of each

LDA actually selected from the LDAPrototype method to the supposedly most suitable LDA from the identical prototype run with respect to the values from the ranking. For comparison, the distribution of the distances for random selection of the prototype is given besides. The corresponding values were generated by simulation with permutation of the ranking. The ECDFs in Figure 2b show the relative number of LDAs, in each of the $P = 500$ prototype runs, that according to the ranking would represent a better choice as prototype. Finally, the bar charts in Figure 2c show the corresponding distribution of the absolute numbers of available better LDAs in the same run in accordance to the determined ranking of prototypes. In addition, simulated 95% confidence intervals for frequencies realized by the use of random selection are also shown.

For $K = 20$, many randomly selected LDAs have a rather large distance of about 0.07 at a total mean value of just below 0.04, while the presented method realizes distances that are on average below 0.01. For increasing K the distances seem to increase as well. While the random selection produces an almost unchanging distribution over an extended range, the distribution of LDAPrototype shifts towards zero. Higher values become less frequent. The ECDFs look very similar for all K, whereby for $K = 35$ slightly lower values are observed for small proportions. This is supported by the only major difference in the bar charts. Modeling 20 or 50 topics, for 50% of the prototype runs there is no better available LDA to choose, while for the modeling of 35 topics this scenario applies for just over 40%. The corresponding confidence intervals in Figure 2c are lowered as well. This is an indication that for $K = 35$ it is easier to find a result that is stable to a certain extent for the basic LDA. This is supported by the fact that the distribution of distances in Figure 2a does not seem to suffer.

5 Discussion

We show that the LDAPrototype method significantly improves the reliability of LDA results compared to a random selection. The presented method has several advantages, e.g. the automated computability, as no need of manual coding procedures. In addition, besides the intuitive statistical approach, the proposed method preserves all components of an LDA model, especially the specific topic assignments of each token in the texts. This means that all analyses previously carried out on individual runs can be applied to the LDAPrototype as well. The results suggest that $K = 35$ topics produces more stable results and might therefore be a more appropriate choice for the number of topics than $K = 20$ or 50 on the given corpus. Further studies to analyze the observed differences in the number of better LDAs as well as the distances to the best LDA between different choices of the numbers of topics, may lead to progress in the field of hyperparameter tuning for the LDA.

References

1. Agrawal, A., Fu, W., Menzies, T.: What is wrong with topic modeling? And how to fix it using search-based software engineering. Information and Software Tech-

Reproduced with permission from Springer Nature

nology **98**, 74–88 (2018). https://doi.org/10.1016/j.infsof.2018.02.005
2. Blei, D.M., Lafferty, J.D.: A Correlated Topic Model of Science. The Annals of Applied Statistics **1**(1), 17–35 (2007). https://doi.org/10.1214/07-AOAS114
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Learning Research **3**, 993–1022 (2003). https://doi.org/10.1162/jmlr.2003.3.4-5.993
4. Chang, J.: lda: Collapsed Gibbs Sampling Methods for Topic Models (2015), https://CRAN.R-project.org/package=lda, R package version 1.4.2
5. Chang, J., Boyd-Graber, J., Gerrish, S., Wang, C., Blei, D.M.: Reading Tea Leaves: How Humans Interpret Topic Models. In: Proceedings of the 22nd International NIPS-Conference. pp. 288–296. Curran Associates Inc. (2009)
6. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proceedings of the National Academy of Sciences **101**(suppl 1), 5228–5235 (2004). https://doi.org/10.1073/pnas.0307752101
7. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference and Prediction. Springer Series in Statistics, Springer New York, 2 edn. (2009). https://doi.org/10.1007/978-0-387-84858-7
8. Jaccard, P.: The distribution of the flora in the alpine zone. New Phytologist **11**(2), 37–50 (1912). https://doi.org/10.1111/j.1469-8137.1912.tb05611.x
9. Kampstra, P.: Beanplot: A Boxplot Alternative for Visual Comparison of Distributions. Journal of Statistical Software, Code Snippets **28**(1), 1–9 (2008). https://doi.org/10.18637/jss.v028.c01
10. Koppers, L., Rieger, J., Boczek, K., von Nordheim, G.: tosca: Tools for Statistical Content Analysis (2019). https://doi.org/10.5281/zenodo.3591068, R package version 0.1-5
11. Maier, D., Waldherr, A., Miltner, P., Wiedemann, G., Niekler, A., Keinert, A., Pfetsch, B., Heyer, G., Reber, U., Häussler, T., Schmid-Petri, H., Adam, S.: Applying LDA Topic Modeling in Communication Research: Toward a Valid and Reliable Methodology. Communication Methods and Measures **12**(2-3), 93–118 (2018). https://doi.org/10.1080/19312458.2018.1430754
12. Mimno, D., Wallach, H.M., Talley, E., Leenders, M., McCallum, A.: Optimizing Semantic Coherence in Topic Models. In: Proceedings of the 2011 EMNLP-Conference. pp. 262—-272. ACL (2011)
13. Nguyen, V.A., Boyd-Graber, J., Resnik, P.: Sometimes Average is Best: The Importance of Averaging for Prediction using MCMC Inference in Topic Modeling. In: Proceedings of the 2014 EMNLP-Conference. pp. 1752–1757. ACL (2014). https://doi.org/10.3115/v1/D14-1182
14. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2019), http://www.R-project.org/
15. Rieger, J.: ldaPrototype: Prototype of Multiple Latent Dirichlet Allocation Runs (2020). https://doi.org/10.5281/zenodo.3604359, R package version 0.1.1
16. Rieger, J., Koppers, L., Jentsch, C., Rahnenührer, J.: Improving Reliability of Latent Dirichlet Allocation by Assessing Its Stability Using Clustering Techniques on Replicated Runs (2020)
17. Roberts, M.E., Stewart, B.M., Tingley, D., Airoldi, E.M.: The Structural Topic Model and Applied Social Science. In: NIPS-Workshop on Topic Models: Computation, Application, and Evaluation (2013)
18. Rosen-Zvi, M., Griffiths, T., Steyvers, M., Smyth, P.: The Author-Topic Model for Authors and Documents. In: Proceedings of the 20th UAI-Conference. pp. 487–494. AUAI Press (2004)