# 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 $\alpha, \eta$ 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 $\alpha$, $\eta$ 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

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

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, $\chi^2$ 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 $\chi^2$ 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 $\chi^2$ 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

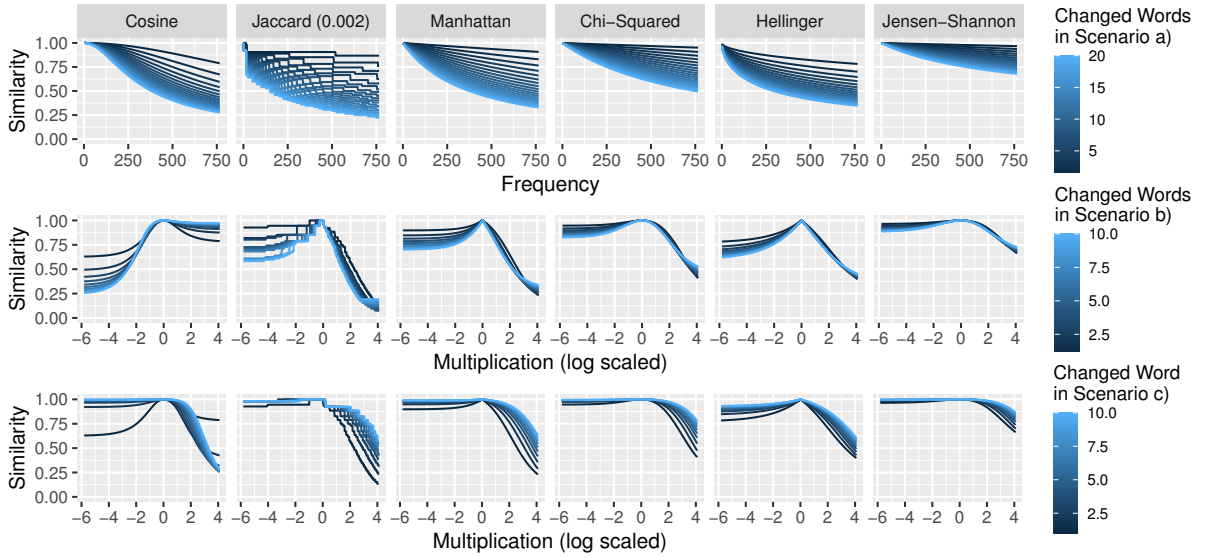


Figure 1: Comparison of similarity measures regarding the effect on topic-self-similarity obtained by modifications of the original word frequency vector with respect to scenario a), b) and c).

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 $\chi^2$ 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

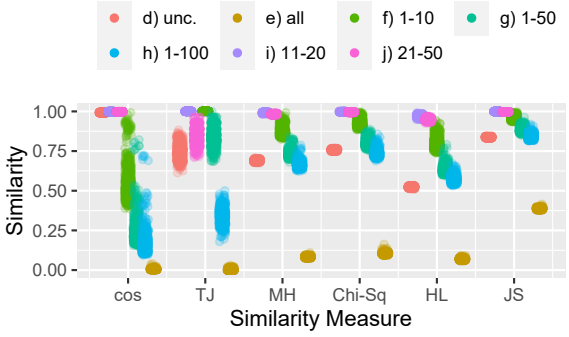


Figure 2: Comparison of similarity measures concerning the effect on self-similarity obtained by applying scenarios d) to j) to the original word frequency vector.

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.

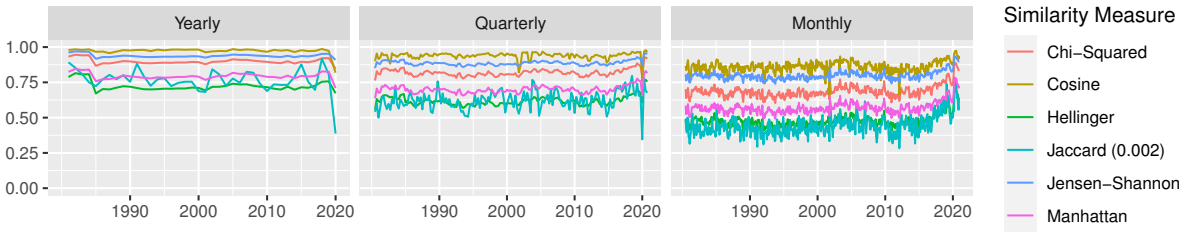


Figure 3: Unit-to-unit self-similarities of the topic *Health* in the scenario of quarterly modeling with three quarters memory and starting with the rolling approach in 1985 for the six different similarity measures.

### 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, 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.

| start | mem-ory | non-changing | | changing | |
|---|---|---|---|---|---|
| | | quarter | year | quarter | 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.

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

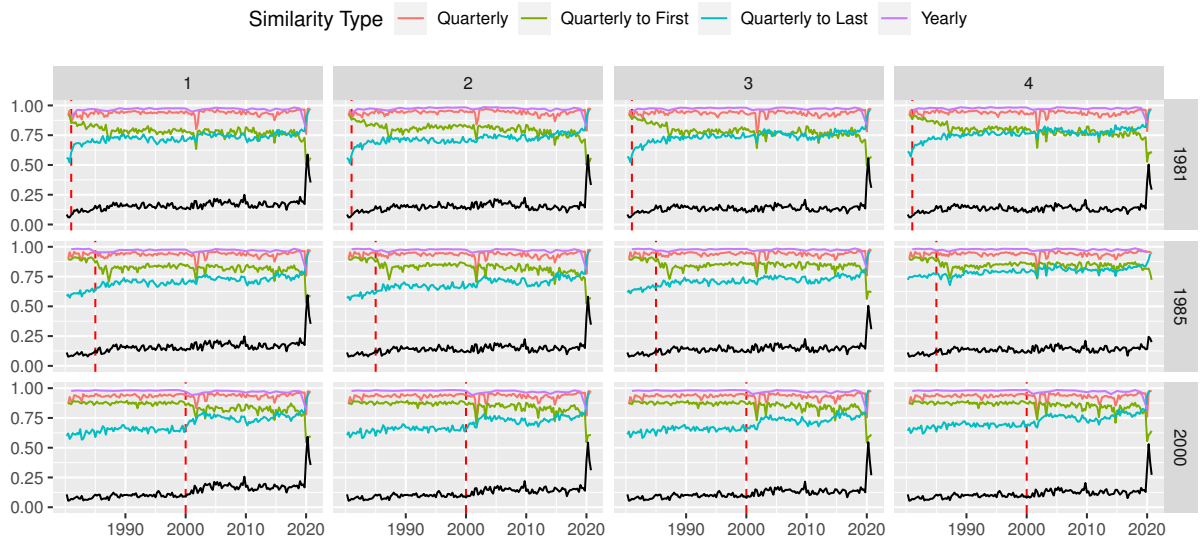


Figure 4: Cosine self-similarities of the topic *Health* for all parameter combinations of the memory and the rolling starting date in the quarterly modeling scenario (topic's scaled share is multiplied by 7 and visualized in *black*).

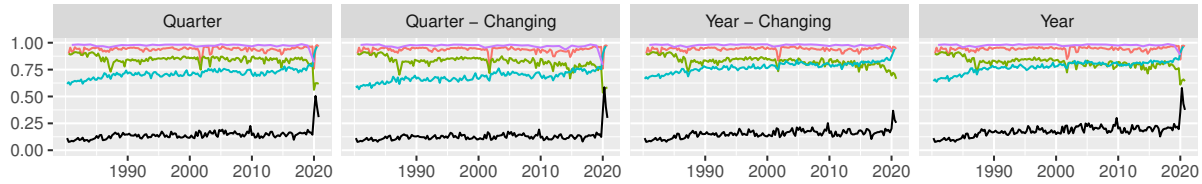


Figure 5: Cosine self-similarities and scaled share of the topic *Health* for *non-changing* and *changing* previous assignments. The scenario of *quarterly* and *yearly* modeling with three quarters memory and starting with the rolling approach in 1985 is considered.

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

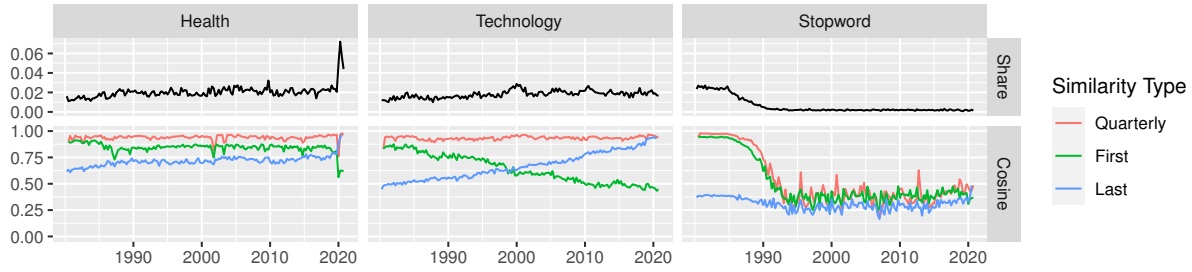


Figure 6: Different patterns of topic's cosine self-similarity for the topics *Health*, *Technology* and a *Stopword* topic in the scenario of quarterly modeling with three quarters memory and starting with the rolling approach in 1985. In addition, the quarterly share of the respective topic is shown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