Christian Pölitz

# Data Mining Software for Corpus Linguistics with Application in Diachronic Linguistics

**Abstract**

Large digital copora have become a valuable resource for linguistic research. We introduce a software tool to efficiently perform Data Mining tasks for diachronic linguistics to investigate linguistic phenomena with respect to time. As a running example, we show a topic model that extracts different meanings from large digital copora over time.

## 1 Introduction

From the 1960s on, modern digital text corpora offer large text collections like newspaper articles, social media content, but also language reference corpora for linguistic analysis. With the Internet, even more textual information has become available for everybody. To use such large amounts of digital texts, non-manual methods to extract information for linguistic research must be used. Data Mining methods, see Manning and Schütze (1999) for example, can help to automatically analyze such large document collections and corpora. Data Mining methods try to discover knowledge from data sources and perform automatic analysis tasks based on identification of patterns in the data. The goal is to find information in the data when manual analyses are not possible, too expensive or too time-consuming.

To demonstrate the need for Data Mining in corpus linguistics, we investigate large digital corpora for temporal dynamics: We show the development of meanings over time. The results will show how useful Data Mining methods can be for such linguistic tasks. In this paper, we propose a software tool for meaning extraction that systematically extends standard approaches to explicitly adopt to corpora from heterogeneous language resources with information about time.

For example, from the Dictionary of the German Language, see Geyken (2007), we can retrieve KWIC[1]-lists of snippets containing the German word *Platte*. The snippets are drawn from documents from different genres over a time period from 1900 to 1999. For these documents, we extract different meaning of the word *Platte* by Latent Dirichlet Allocation (Blei et al. (2003)). In Figure 1, we illustrate two extracted meanings. At the top, we show the words that are most important for each meaning by a Word Cloud[2]. We see that we identify two different meanings of the word *Platte*. First, *Platte* in the meaning of a hard drive is found. The most important words are highly computer related. The second meaning identifies the word *Platte* as photographic plate. The important words are all connected to photography.

[1] Key-word-in-context: Examples of word (or expression) usage in texts.
[2] A Word Cloud visualizes frequent words and their importance by font size.

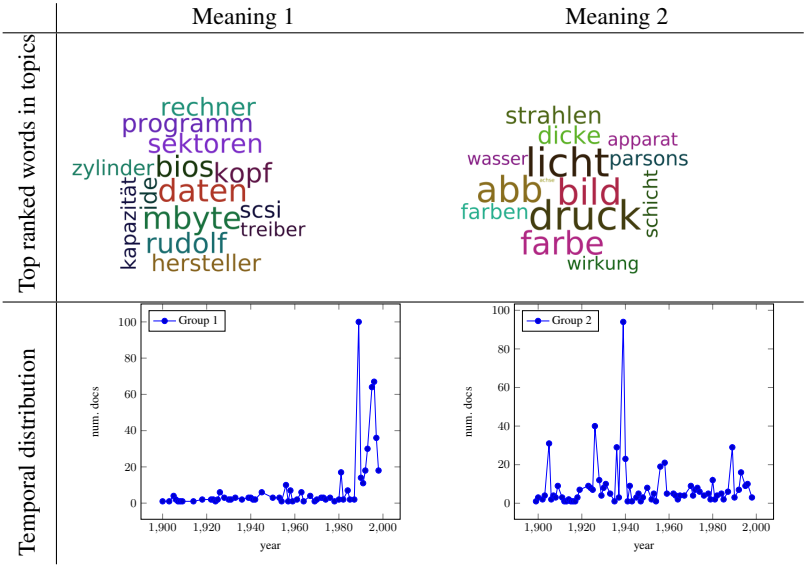


**Figure 1:** Two possible meanings for the word *Platte* (plate) extracted from KWIC-lists of snippets from the Core Corpus of the Dictionary of the German Language. Top: The most important words in the topics. Bottom: The temporal distribution of the meanings.

At the bottom, we plot the amount of the usage of the different meanings over the time: We count how many documents are assigned to the meanings in each year. We see that *Platte* in the context of a hard drive is mostly used at the end of the 20th century, while *Platte* in the context of photography is mainly used in the 1950s.

Based on this first study. we developed a software tool to implement different versions of topic models and topic models with temporal information that can be used for diachronic linguistics. Besides, algorithms for topic modeling, our software tool offers methods to evaluate and visualize the results on large digital corpora.

## 2 Topic Models

Topic models are statistical models that extract semantics in document collections based on co-occurrence statistics. For these models, we assume the Multinomial Model (MM): The words in the documents are drawn from Multinomial distributions. The most prominent latent topic models are Probabilistic Latent Semantic Analysis (Deerwester et al. (1990)) and Latent Dirichlet Allocation (Blei et al. (2003)). Both models are mixture models (McLachlan and Basford (1988)) that model the joint probability of words and documents as linear combination of conditional distribution of the latent topics.

### 2.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) as proposed by Blei et al. (2003) is a generative probabilistic topic model that estimates document-topic distributions $\theta$ and topic-word distributions $\beta$ with Dirichlet priors $\mathrm{Dir}(.)$. Given a corpus $C$ of $m$ documents, each represented by a sequence of words $\mathbf{d} = (w_1, \cdots, w_n)$, LDA models the generative process of generating documents as random draws over random mixtures of latent topics $t$. We briefly summaries the generative process of documents as the following:

1. For each topic t:
   a) Draw $\beta_t \sim \mathrm{Dir}(\eta)$
2. For each document $d \in C$:
   a) Draw $\theta_d \sim \mathrm{Dir}(\alpha)$
   b) For each word i:
      i. Draw $t_i \sim \mathrm{Mult}(\theta_d)$
      ii. Draw $w_i \sim \mathrm{Mult}(\beta_{t_i})$

First, we draw for each topic $t$ the word probabilities $\beta_t$ for each word in the corpus. Next, for each document $d$ we draw a $T$-dimensional Dirichlet distributed random vector $\theta_d$. Then, for each word in the document $d$ we draw a topic $t_i$ from a Multinomial distribution parametrized by $\theta_d$ and a word $w_i$ from a Multinomial distribution parametrized by $\beta_{t_i}$. In the original approach by Blei et al., $\beta_t$ does not have a Dirichlet prior $Dir(\eta)$. This is important for sampling based approaches for LDA and for possible extensions with different (more complicated) priors.

In the literature there are two major approaches to estimate an LDA topic model. First, variational inference can be used to approximate the posterior distribution of the latent variables by a simpler variational distribution (Blei et al. (2003)). Second, Gibbs sampling defines a sequence of random draws that converges to a sequence of topic assignments that follows the joint distribution of the topic model (Griffiths and Steyvers (2004)). In the software tool, we implemented both approaches.

## 3 Temporal Topic Models

While the standard topic models group only words and documents in semantically related topics, we are further interested in the distribution of the topics over time. Certain meanings of words, for example, might be used only in certain time periods: The word *cloud* for instance has recently become a new meaning of a "data cloud". Further, there can be certain trends or attentions to topics. Topics about US presidents for example will very likely be highly present around a year of elections.

In order to extract the distribution of topics over time, we use topic models that consider temporal information about the documents. Each document has a time stamp $\tau$. We assume that each word in the documents is associated with this time stamp. The time stamps follow the

distribution $p(\tau|m)$ for meta parameters $m$ and are assumed to be conditionally independent given a topic. Hence, we model the time stamps as additional observed random variables that depend on the topics.

A specific instance of a probability distributions of the time stamps is the Beta distribution. Wang and McCallum (2006) introduced this model to investigate topics over time. They call this method Topics over Time (TOT). The generative process of standard LDA is extended such that for each word $w_i$ in each document, we also draw a time stamp $\tau_i \sim \text{Beta}(m_{t_i})$ with $m_{t_i} = (a, b)$ the shape parameters of the Beta distribution for topic $t_i$.

In the software tool, we provide implementations for several distributions. The parameters of the distributions are additionally estimated by Maximum Likelihood Estimation using Newton-like gradient descent with a standard BFGS optimization solver, see Liu and Nocedal (1989).

## 4 Evaluation

We implemented several standard evaluation methods for topic models. Beside coherence measures that estimate the quality of a topic based on external knowledge of word correlations, we also provide methods to estimate likelihoods of test document collections. To qualitatively evaluate topics, we provide statistics to visualize the results of a topic model.

### 4.1 Coherence Measures

Frequently used quantitative evaluation methods are based on the relations of the highest ranked words in each topic. The coherence measures estimate how well the model fits an as coherent expected outcome. The definition of this expected coherent outcome is usually based on user studies and experience with topic modeling in practice. A fundamental assumption for topics or factors to be coherent is based on the top ranked words. Each topic is associated with a value how present this topic is for given words. This value can be directly read of from the multinomial distribution $\beta_t$. Ranking the words for each topic results in a compact representation of the each topic.

For $T$ latent variables with corresponding top-$k$ words in ranking lists $V_t = \{w_{1t}, \cdots, w_{kt}\}$ with respect to each latent variable that is extracted by a latent variable model, the overall coherence measure is the mean over individual coherence values $U(V_t)$:

$$U(V) = \frac{1}{T} \sum_{t=1}^{T} U(V_t).$$

To estimate the individual values for a given latent variable model, we use several coherence measures that have been proposed in the literature. All measures use statistics of co-occurring words from an additionally given reference document collections like Wikipedia articles. For a detailed description of the quality measures see Röder et al. (2015). In the next subsections, we describe the coherence measures mostly used in literature for topic models.

#### 4.1.1 UMass

In Mimno et al. (2011), the authors propose a topic coherence measure that depends on co-occurrences of words. Based on user studies, they show that this measure corresponds well with the top ranked topics by the users. In the literature the measure is called the $U_{Mass}$ measure and is defined as

$$U_{Mass}(V_t) = \sum_{m=2}^{k} \sum_{l=1}^{m} \log \frac{D(w_{mt}, w_{lt}) + 1}{D(w_{lt})}. \tag{1}$$

The measure is the sum of the log-ratios of the by 1-smoothed co-occurrence frequency of any two ordered words in the top ranked list, $D(w_{mt}, w_{lt})$, and the document frequency of the lower ranked word, $D(w_{lt})$.

#### 4.1.2 Pointwise Mutual Information

The authors in Newman et al. (2010) introduce Pointwise Mututal Information (PMI) as measure for topic coherence. The PMI is the log-ratio of the joint probability of two random variables and the product of their marginal probabilities. It measures how likely two random variables are jointly distributed and not independently distributed. The PMI of two words $w_1$ and $w_2$ is defined as the following:

$$PMI(w_1, w_2) = \log \frac{p(w_1, w_2)}{p(w_1)p(w_2)}.$$

The PMI can be interpreted as how much likely the two words $w_1$ and $w_2$ appear together in contrast to how likely they appear alone.

For a latent topic, respectively factor $t$ and the top-$k$ ranked words $V_t$, the PMI is defined as:

$$PMI(V_t) = \frac{1}{(k-1)k/2} \sum_{m<n}^{k} \log \frac{p(w_{mt}, w_{nt})}{p(w_{mt})p(w_{nt})}. \tag{2}$$

In Aletras and Stevenson (2013) propose to use the Normalized Pointwise Mututal Information (NPMI) to estimate the coherence of the topics. The NPMI is the PMI divided by the negative log probability of the two words appearing together. The reason to use NPMI is twofold. First, NPMI is normalized between $-1$ and 1. Second, low frequencies of the words are less critical. Especially the second reason is important, since small outliers can result in very small joint probabilities that overtake the whole coherence measure. Formally the NPMI is defined as:

$$NPMI(V_t) = \frac{1}{(k-1)k/2} \sum_{m<n} \frac{\log \frac{p(w_{mt}, w_{nt})}{p(w_{mt})p(w_{nt})}}{-\log p(w_{mt}, w_{nt})}. \tag{3}$$

In our software tool, we provide these coherence measures via an interface to the library Palmetto (Röder et al. (2015)). The estimations of the frequencies and probabilities are all based on word indices from Wikipedia corpora (German and English). To generate these indices, the

Wikipedia corpora must be retrieved (for example from the Institute of the German Language[3]) and a Lucene-based index must created as explained here: `https://github.com/AKSW/Palmetto/wiki/How-to-create-a-new-index`.

#### 4.1.3 Temporal Coherence for Temporal Topic Models

Similar to the coherence of the top ranked words, we estimate the temporal coherence as distance of the distribution of the time stamps associated with a latent topic, with the distribution of the time stamps for the top words over all documents in the corpus. We assume that the documents containing the top words from latent variables approximate the content of the underlying concept. The temporal difference of the time stamps of these documents indicates how well this latent information captures the true temporal dynamics in the corpus. A topic is temporal coherent if the estimated distribution of the time stamps in this topic is similar to the temporal distribution of the time stamps for the top words in the whole corpus. The documents that contain the top two words approximate the semantics behind the topic. Hence, documents containing the top two words in the corpus can be used as coherence reference.

The empirical distributions of the time stamps of the topics and the top words in the corpus are estimated by histograms $h_t$ and $h_{w_1,w_2}$. For a topic $t$, the empirical probability of the time between two time stamps $\tau_1$ and $\tau_2$ can be approximated by

$$P([\tau_1, \tau_2]|t) = p(\tau_2|t) - p(\tau_1|t) \propto \sum_{\tau} I_{\tau_1 \leq \tau < \tau_2}(\tau) n_{\tau,t}.$$

For $n_{\tau,t}$ the number of tokens assigned to topic $t$ from a document with time stamp $\tau$ and the indicator function

$$I_{\tau_1 \leq \tau < \tau_2}(\tau) = \begin{cases} 1, & \tau \in [\tau_1, \tau_2] \\ 0, & else \end{cases} .$$

Now, we define the histogram of the temporal distribution of topic $t$ as function $h_t : \mathbb{N} \to \mathbb{N}$ such that

$$h_t(\tau_1, \tau_2) = \sum_{\tau} I_{\tau_i \leq \tau < \tau_{i+1}}(\tau) n_{\tau,t},$$

for a given number of intervals $[\tau_1, \tau_2], \cdots, [\tau_{e-1}, \tau_e]$.

For two words $w_1$ and $w_2$ for topic $t$, the empirical probability can be approximated by

$$P([\tau_1, \tau_2]|w_1, w_2) = p(\tau_2|w_1, w_2) - p(\tau_1|w_1, w_2) \propto \sum_{\tau} I_{\tau_1 \leq \tau < \tau_2}(\tau) n_{w_1,w_2,\tau}$$

for $n_{w_1,w_2,\tau}$ the number of tokens in the documents that contain both words $w_1$ and $w_2$ in the corpus with time stamp $\tau$. The histogram of the temporal distribution of the words $w_1$ and $w_2$ in

[3] `http://www1.ids-mannheim.de/kl/projekte/korpora/verfuegbarkeit.html`

the corpus is the function $h_{w_1,w_2} : \mathbb{N} \to \mathbb{N}$ such that

$$h_{w_1,w_2}(\tau_1, \tau_2) = \sum_{\tau} I_{\tau_1 \leq \tau < \tau_2}(\tau) n_{w_1,w_2,\tau}.$$

There are several distance measures possible. We propose to use the Minkowski distance to estimate how much the distributions over the time stamps differ based on histograms. The Minkowski distance of two histograms for topic $t$ and the corresponding top two words $w_{1t}, w_{2t}$ is defined as

$$D(h_t, h_{w_{1_t},w_{2_t}}, p) = \sqrt[p]{\sum_i |h_t(\tau_i, \tau_{i+1}) - h_{w_{1_t},w_{2_t}}(\tau_i, \tau_{i+1})|^p}.$$

Using $p = 2$ is the Euclidean distances and $p = 1$ is the $l_1$ distance.

### 4.2 Likelihood

The coherence measures estimate the quality of latent topics based on statistics from different document collections and user information. To estimate how good a factor or topic model fits the corpus we estimate the likelihood of the data under this model. Depending on the specific model, we can directly estimate the likelihood or we need special assumptions. For topic models, the likelihood of a set of test documents in corpus $C_{te}$ given a topic model by its parameters is

$$p(C_{te}|\alpha, \beta) = \prod_{d \in C_{te}} p(\mathbf{d}|\alpha, \beta).$$

#### 4.2.1 Sequential Monte Carlo

As proposed by Wallach in Wallach et al. (2009), Sequential Monte Carlo Method can be used to estimate the likelihood of a topic model. For a sequence of words from a hold-out test data set, the probability of the test words $w$ is

$$p(\mathbf{d}|\alpha, \beta) = \prod_m p(w_m|\mathbf{d}_{<m}, \alpha, \beta).$$

A Sequential Monte Carlo algorithms to estimate the likelihood of a held-out data set for a given topic model can be defined in the following way: Given a new document $d$ as sequence of tokens, $\mathbf{d} = (w_1, \cdots, w_N)$, we re-sample topic proportions for each token $w_m$ in $\mathbf{d}$, given all tokens before, $\mathbf{d}_{<m} = (w_1, \cdots, w_{m-1})$, using the point estimate of the topic-word distributions. To compensate the uncertainty in these estimates for a single document, we keep $M$ independent samples. These samples are called particles. For the $m_{th}$ word in the sequence, the probability is

$$p(w_m|\mathbf{d}_{<m}, \alpha, \beta) = \sum_{i=1}^{T} p(t_i|\theta) p(w_m|\mathbf{d}_{<m}, t_i, \beta) \tag{4}$$

$$= \sum_{i=1}^{T} \frac{n_{d,i,<m} + \alpha_k}{n_i + \sum_{k'} \alpha_{k'}} \beta_{t_i, w_m}. \tag{5}$$

This is a mixture of multinomial Distribution with Dirichlet prior $\mathrm{Dir}(\eta)$, with mixing weights $p(t_i|\theta)$ for Dirichlet ($\mathrm{Dir}(\alpha)$) distributed $p(t|\theta)$.

We apply Sequential Monte Carlo Methods using particle learning (PL) methods as proposed by Scott and Baldridge (2013) and by Naesseth et al. (2014). To get an estimate for the topic weights, we use aggregated counts of topic assignments for topics $i$: $n_i$, respectively for the document $d$: $n_{d,i}$. For $m = 1, \cdots, M$, we use aggregated counts $n_{d,i,<m}$, with count assignments for all tokens up to the $m_{th}$, sampled iteratively from

$$p(t = i|w_m, t_{<m}) \propto \frac{\alpha_k}{\sum_{k'} \alpha_{k'}} \beta_{m,i}$$

and collected as particles. We re-sample for topic proportions for the documents, but use the point estimate for the word distribution in each topic. Then, we sample for each particle and its corresponding aggregated counts, topic assignments and add them to these counts. This means, we have $Z$ estimates of the aggregated counts and consequently can estimate $Z$ times $p(w_m)$. This models the uncertainty about the assignment by $Z$ particles.

We define particles $T_{m,z} \sim p(t|w_1, \cdots, w_m, \tau_d, \beta)$ for $z = 1, \cdots, Z$. The $T_{m,z}$ are iteratively sampled such that $T_{N,z} \sim p(t|\mathbf{d}, \tau_d, \beta)$.

For temporal topic models by additional random variables that depend on the latent topics, we can easily extend to Sequential Monte Carlo method from above to estimate the likelihood of hold-out documents with additional time stamps:

$$p(w_m, \tau_d|\mathbf{d}_{<m}, \alpha, \beta) = \sum_{i=1}^{T} p(t_i|\theta) p(w_m, \tau_d|\mathbf{d}_{<m}, t_i, \beta) \tag{6}$$

$$= \sum_{i=1}^{T} \frac{n_{d,i,<m} + \alpha_k}{n_i + \sum_{k'} \alpha_{k'}} \beta_{t_i, w_m} p(\tau_d|t_i). \tag{7}$$

This is a mixture of multinomial distributions with Dirichlet prior $\mathrm{Dir}(\eta)$, with mixing weights $p(t_i|\theta)p(\tau_d|t_i)$ for Dirichlet ($\mathrm{Dir}(\alpha)$) distributed $p(t|\theta)$ and Shifted-Gompertz distributed $p(\tau|t)$. This density is analogue to Equation 4 using additional time stamps. The difference lies in the integration of the temporal distributions. This is easy due to the independence assumption in temporal topic modeling.

Besides the joint likelihood of the words and the time stamps, we are also interested in the conditional likelihood. The conditional likelihood is the likelihood of the sequence of words in a test document given the time stamp: $p(\mathbf{d}|\tau_d)$. This conditional likelihood estimates the likelihood of words from the documents at the time of the document. This measure focuses on the quality of the estimated word distribution. Due to the independence assumption in the topic models, we have the following conditional probability of a sequence of words in a document given the corresponding time stamp:

$$p(\mathbf{d}|\tau_d) = \prod_n p(w_n|\tau_d).$$

The partial conditional probabilities can be calculated via

$$p(w_n|\tau_d) = \frac{p(w_n, \tau_d)}{p(\tau_d)}.$$

The joint probability $p(w_n, \tau_d)$ is estimated as in Equations 5 and the probability of the time stamp $\tau_d$ is

$$p(\tau_d) = \sum_t p(\tau_d|t).$$

### 4.3 Qualitative Evaluation

In practice, the topic models are used qualitatively. Experts interpret results of the models by exploring the topics. For linguistic tasks for example, the results of latent topic modeling are mostly manually investigated. For example, if we are interested in usage patterns of expression and words in context and over time, we need methods to manually evaluate the results of a latent topic model in terms of the words and documents. Rather, than abstract numbers that describe the results, we are interested in how explanatory the topics are. A good format and a visualisation of the results is needed to help evaluating the models by linguists. There are several possible ways to visualize the results of topic models. In the literature there are usually the following aspects considered: First, how can we show the tendency of words and documents to certain topics. Second, how can we show the distribution of the topics over the words and documents. Finally, how can we show the distribution of topics, words and documents over time. The latter is important for diachronic linguistics.

#### 4.3.1 Word Clouds

A concrete visualization of the words with respect to their importance is a World Cloud. Each of the top-$k$ words from the ranking list is written in a figure with size proportional to $\beta_{t,w}$. On the left in Figure 2, we see a Word Cloud from a topic about US president Obama. Besides Word Clouds, we can list the top-$k$ words in decreasing order of importances (in the concrete values of $\beta_{t,w}$) and additionally plot these values as histogram. This can be seen on the right in Figure 2. In the software tool, we provide the top ranked words from a topic model as CSV[4] result file.

[4] A CSV file contains table like data as comma separated values.

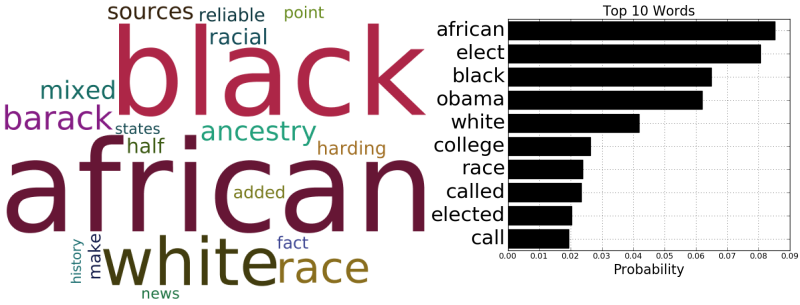


**Figure 2:** Visualization of the most important words for a given topic. Left: a Word Cloud from the highest ranked words from a topic model on Wikipedia talk pages containing the word “president”; right: a sorted list of the highest ranked words.

#### 4.3.2 Temporal Distribution

Ranking lists and Word Clouds can visualize the word distributions for different topics. For the temporal distributions of the documents with respect to the topics, we need to display the course of the importance of the latent topic over time. The amount of a certain topic in a given time can be estimated by grouping documents by time and averaging the document-topic proportions $n_{d,i} \propto \theta$. The document-topic proportion tells how much present a certain topic is in a document.

Each document $d$ has its time stamp $\tau_d$. Grouping these values into $n$ intervals:

$$[0, \tau_1], [\tau_1, \tau_2], \cdots, [\tau_{n-1}, \tau_n],$$

we assign the documents to the corresponding intervals, hence $d \rightarrow [\tau_i, \tau_{i+1}]$ with $\tau_i \leq \tau_d \leq \tau_{i+1}$. Now, we can average $n_{d,i}$ in each interval to get a histogram of topic $i$ over time. Additionally, we can plot the estimated temporal distribution. In Figure 3, we show the density of an estimated distribution of time stamps together with the corresponding histogram of time stamps for a topic from a topic model trained on Wikipedia talk pages about presidents. Similar to the top words, we additionally protocol the time stamps associated with the topics from a temporal topic model and the estimated parameters for the temporal distributions.

## 5 Software

This section describes the plugin “Corpus Linguistic Plugin” as extension for the Data Mining tool RapidMiner. We explain in detail the software and how it can be used in diachronic linguistics. The software is already used in corpus linguistic research and teaching at the TU Dortmund University and the Mannheim University. We start the software description with an introduction to Data Mining tool RapidMiner. RapidMiner is used and extended for corpus linguistic tasks in our software tool. The plugin can be downloaded from: `http://sfb876.tu-dortmund.`

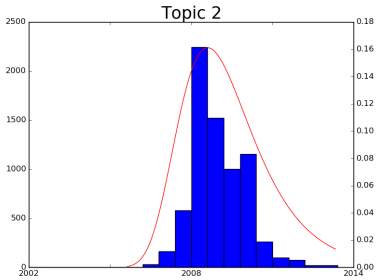


**Figure 3:** Visualization of the distribution of topics over time as plot of a histogram of topic proportions for a given topic over time from a topic model about presidents. Additionally, we plot the density of the time stamps fitting the corresponding time stamps for this topic.

`de/auto?self=Software` under the link: "Corpus Linguistics Plugin" or directly at `http://sfb876.tu-dortmund.de/auto?self=%24es4hb8h0cg`.

### 5.1 RapidMiner

RapidMiner, as for instance described by Hofmann and Klinkenberg (2013), is a Data Mining toolbox used to perform data analysis on different data sources. RapidMiner offers the classical analysis and Data Mining steps from data retrieval to data transformation and pre-processing, performance of analysis and Data Mining methods to evaluation methods, post-processing and visualization. Individual processing steps are performed by so called **Operators**. The standard operators are separated into several categories and are organized in an ontology represented as folder structure in the operator explorer view on the left of the main screen as seen in Figure 4. The main categories of operators are:

- import/export operators: reading and writing of data
- data transformation operators: pre- and post-processing of data
- modeling: analytic and Data Mining methods on data
- evaluation operators: quality estimation of the modeling results.

The operators are compiled to a sequence of steps summarized in a so called **Process**. This process defines a flow of input data to processing operators that output result data. In the middle Figure 4, an example process is shown with the execution order of the individual operators. Starting with reading data as CSV-file, the data is pre-processed by transforming nominal to numeric data. The modeling operator **SVM** builds a classification model that is applied on test data which is additional read in. Finally, the **Performance** operator is used to evaluate the model by standard measures. The operators have a number of parameters to be specified. On the right Figure 4, the **Parameters** panel is shown as input mask for all parameters. Clicking on an operator, this

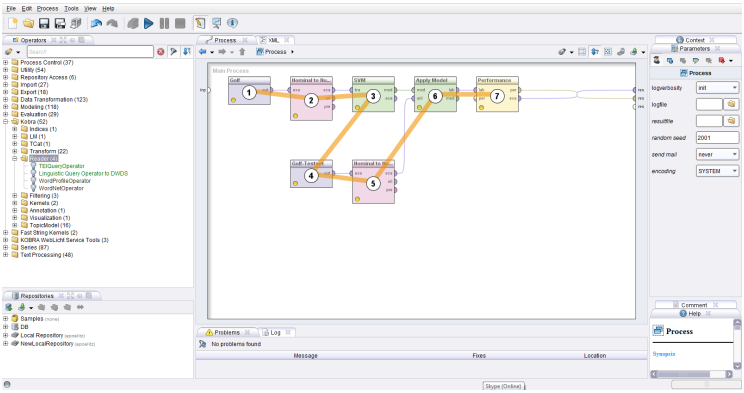

**Figure 4:** RapidMiner user interface. Left: Operators for data loading, pre-processing, Data Mining methods, post-processing and data export. In the middle: Data Mining process. Right: Properties and parameters of the process and the operators.

panel shows the parameters that need to be set for this operator. Additional, a description of each operator can be found on the **Help** panel. A general introduction for Data Mining with RapidMiner can be found in the book by North (2012).

### 5.2 Corpus Linguistics Plugin

RapidMiner offers a convenient interface and a plethora of available analyses methods. Compared to low level interfaces and libraries for different programming languages, RapidMiner offers a more user friendly tool box. This makes the introduction of our methods more easy for linguistic researchers - even with little knowledge in computer science. We implemented different versions of topic models and evaluation methods as a plugin for the RapidMiner. For the different topic models, different operators are available. Besides standard LDA with Gibbs sampling and Variational Inference, supervised versions with Gaussian, Beta, Uniform and Gompertz distributed document labels can be used for diachronic linguistics. Additionally, an implementation of topic models with word features and word groups via special Laplace and Group-Sparsity inducing priors is available to integrate word informations.

To access different corpora, operators to execute linguistic queries on corpora at the Berlin Brandenburger Academia of Science are available. Besides the standard corpora, we also provide access to dictionaries and GermaNet (the German version of WordNet). To access Wikipedia corpora, a TEI-reader is implemented that extends a standard XML-stream reader to process Text Encoding Initiative (TEI) tags, see Beißwenger et al. (2012). Finally, pre-processing and post-processing operators provide methods for text transformations and text visualization. In the next subsections, concrete examples for the use of the plugin are described.