Software

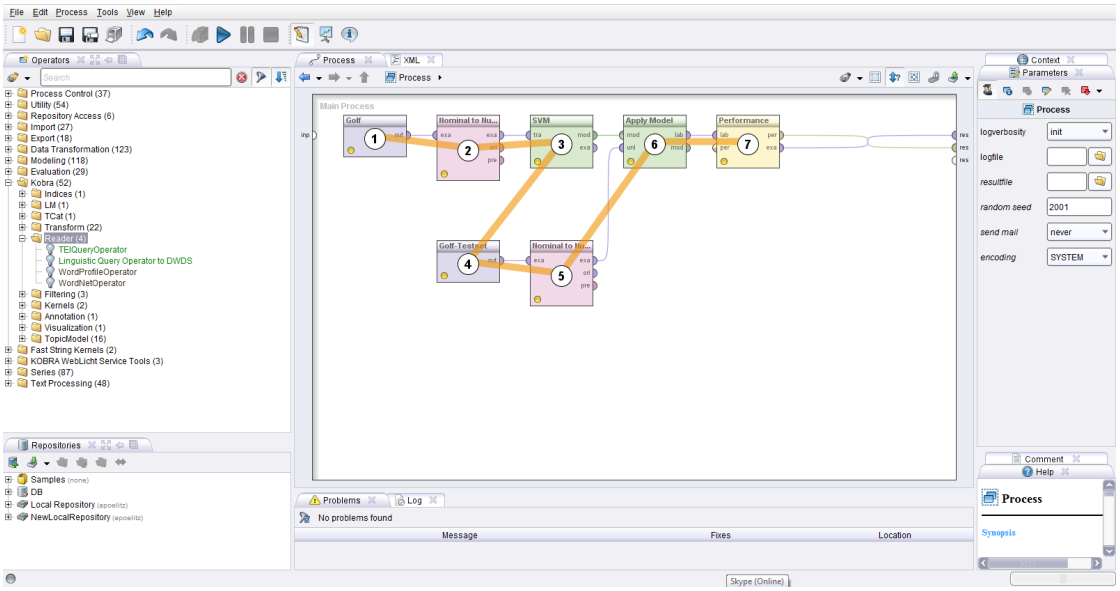


Figure 0.1: RapidMiner user interface:

This paper describes the software plugin "Corpus Linguistic Plugin" for RapdiMiner. We explain in detail the software and how it can be used to produce use cases in variety and diachronic linguistic. 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 the Data Mining tool RapidMiner. The RapidMiner is used and extended for corpus linguistic tasks.

RapidMiner

The RapidMiner [5] 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 0.1. The main categories of operators are:

- import/export operators: reading and writing of data
- data transformation operators: pre- and post-processing of data
- modelling: analytic and data mining methods on data
- evaluation operators: quality estimation of the modelling 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 of the figure, 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 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 of the figure, the **Parameters** panel is shown as input mask for all parameters. Clicking on an operator, this 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 into Data Mining with RapidMiner can be found in the book [7] by Matthew North.

Corpus Linguistic Plugin

The 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 with little knowledge in computer science. We implemented the proposed latent variable methods as a Plugin for the RapidMiner. For the different variants of Latent Dirichlet Allocation (LDA), 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 linguistic tasks. An implementation of LDA with word features and word groups via special Laplace and Group-Sparsity inducing priors is available to integrate word informations. Some of the latent factor methods can be generated with existing operators already available in RapidMiner. For example for Latent Semantic Analysis (LSA), the available operator for a Singular Value Decomposition (SVD) can be used. For variety linguistic tasks, we provide an operator that extract latent factors that match distributions of different document collections.

Additional to the latent variable methods, we also implemented a number of interfaces to the language resources. To access the different corpora, operators to execute linguistic queries on the different corpora at the Berlin Brandenburger Academia of Science are available. Besides the standard corpora, we also provide access to the dictionaries and the GermaNet (the German version of WordNet). To access the Wikipedia corpora, a TEI-reader is implemented thats extends a standard XML-stream reader to process the Text Encoding Initiative (TEI) tags, see [1]. Finally, preprocessing operators provide methods for text transformations and text visualization. In the next subsections, concrete examples for the use of the Plugin are described. A reference for the individual operators is given in the appendix.

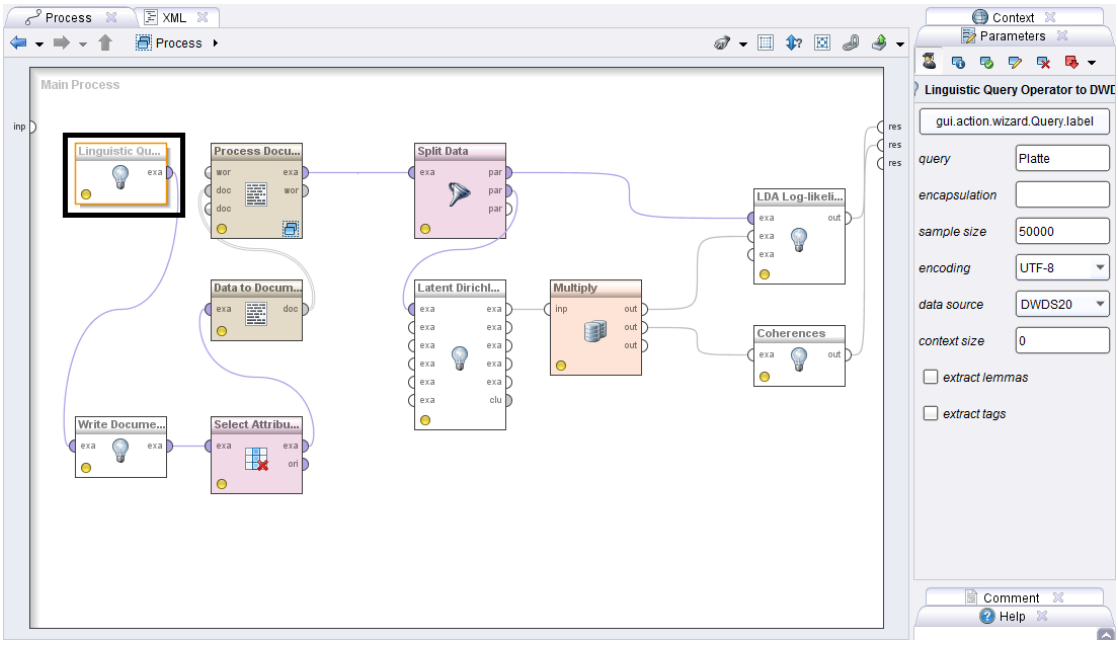


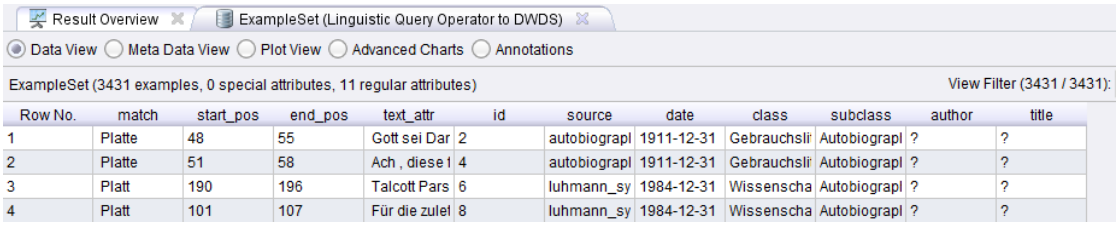
Figure 0.2: Linguistic Query Operator as first step in a process to perform a linguistic task by latent variable methods. For a given query, we retrieve KWIC-lists from a corpus.

Interface to Linguistic Resources

The first step to perform linguistic tasks with the Corpus Linguistic Plugin is the retrieval of the data. The KWIC-lists or documents are extracted and internally represented as string. Standard text documents can be opened by the **Read CSV** operator from RapidMiner. For the linguistic corpora we implemented the Linguistic Query Operator as shown in Figure 0.2. The **Linguistic Query Operator** provides access to the DWDS Core-Corpus of the 20th century, the Core Corpus of the German text archive and the Die Zeit corpus of news articles from 1947 to 2014. For a given linguistic query, the operator retrieves a number of concordances and generates an example set that contains the texts, a time stamp and additional information about author and source. The query is sent to a server at the Berlin Brandenburger Academia of Science and a Perl script runs the query against a Dialing and DWDS Concordance (DDC) data base containing the corpora, see [9]. The KWIC-lists are returned as JSON[1] files and the operator parses these information and generates the results. Depending on the corpus additional information about the genre of the corresponding documents are also available. Additionally, the position of the query match in the retrieved snippet is given to efficiently identify to match. In Figure 0.4, we show the resulting example set from the Linguistic Query Operator.

For corpora and documents in TEI format, the **TEI Query Operator** provides a

[1] http://www.json.org/



Row No.	match	start_pos	end_pos	text_attr	id	source	date	class	subclass	author	title
1	Platte	48	55	Gott sei Dar	2	autobiograpl	1911-12-31	Gebrauchsli	Autobiograpl	?	?
2	Platte	51	58	Ach , diese l	4	autobiograpl	1911-12-31	Gebrauchsli	Autobiograpl	?	?
3	Platt	190	196	Talcott Pars	6	luhmann_sy	1984-12-31	Wissenscha	Autobiograpl	?	?
4	Platt	101	107	Für die zulet	8	luhmann_sy	1984-12-31	Wissenscha	Autobiograpl	?	?

Figure 0.3: Result example set from Linguistic Query Operator for a linguistic query. For each match of the query, we have an example with information about the match.

stream reader to process large files. Since these files are not indexed as the corpora from the Dictionary of the German Language, we cannot pose linguistic queries. Instead, standard regular expressions can be queried. For the main TEI formatted corpora, the Wikipedia articles and talk pages, the operator retrieves matches of the regular expressions on sentence, paragraph or postings level. These levels are semi-automatic annotated, see [6]. The operator itself implements an XML based stream reader to iterate over the elements from the TEI file. The resulting example set has the same schema as the example set from the Linguistic Query Operator.

To efficiently inspect the retrieved KWIC-lists, the **Annotation Operator** visualizes the text snippets and highlights the matches in the texts. We can also add additional labels or attributes to the texts to further annotated them. The operator generates a result as example set containing the texts of the snippets and the additional annotations.

For the retrieval of information from the additional language resources like dictionaries and WordNets, we implemented operators that can extract these information from local files (WordNet for instance) and retrieve them from the Dictionary of the German Language. The **WordNet Operator** takes a word as parameter and extracts similar words from an existing WordNet instance, given the path to the index, hyponyms and hypernyms for the data. The **GermaNet Operator** works the same, but uses the GermaNet source provided by the Seminar für Sprachwissenschaften at University Tübingen. The retrieval of these information is done by a web service at the Berlin Brandenburg Academia of Science via JSON files. The resulting example set contains for the word of interest given as parameter, the hyponyms and hyperonyms with additional examples and descriptions. Further, the **WordProfiles Operator** retrieves the word profiles provided by the Dictionary of the German Language. These profiles contain words that co-occur with a given word of interest and gives information about the relation between them, see [4].

Text Processing

Before we can use the KWIC-lists for latent variable methods, we need to generate Word-Vectors. With the **Text Processing Plugin** as provided by RapidMiner, text (general strings) can be transformed into a Bag-of-Words and Word-Vectors. Given text in an example set, an internal data structure to represent the text is a generated. This

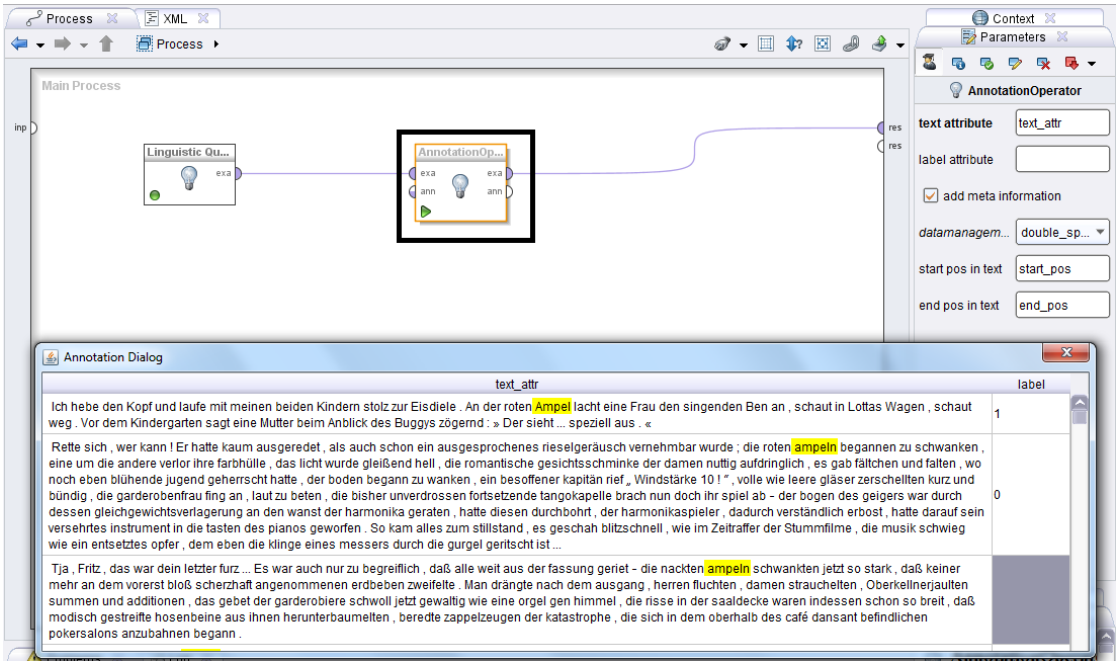


Figure 0.4: Annotation Operator and annotation environment. Given an example set from TEI or Linguistic Query Operator the snippets are visualized and the match is highlighted to inspect the results. Additional annotation can be added like a label.

data structure is called a **Document**. These Documents are further transformed into Word-Vectors containing word occurrences and possible weights as TF-IDF. The text processing plugin offers additional methods to tokenize text by regular expressions or identification of words. Filtering operators can be used to filter out stop words, large tokens or tokens with no characters. Additional pruning mechanisms can be used to filter out words that appear in too many or too few documents. In Figure 0.5, we show how the snippets are transformed into Documents by the **Data to Documents Operator** and how we further generate Word-Vectors by the **Process Documents Operator**. Using the Process Document Operator, we can use different tokenizers to separate the text into tokens and prune words. The resulting example set contains each Document as Word-Vector in a table. In the Word-Vectors there can be pure occurrence information or weighted values like TF or TF-IDF values. We can choose between different methods to prune words. We can prune words with a frequency high or lower threshold, that appear more often or less than a given number or are below and higher a given rank.

Latent Topic Models

From the Bag-of-Words representation of the documents, we can use the sequences of word tokens for the extraction of topics by LDA. Our **Latent Dirichlet Allocation** operator takes the texts as Word-Vector with pure word occurrences and extracts latent

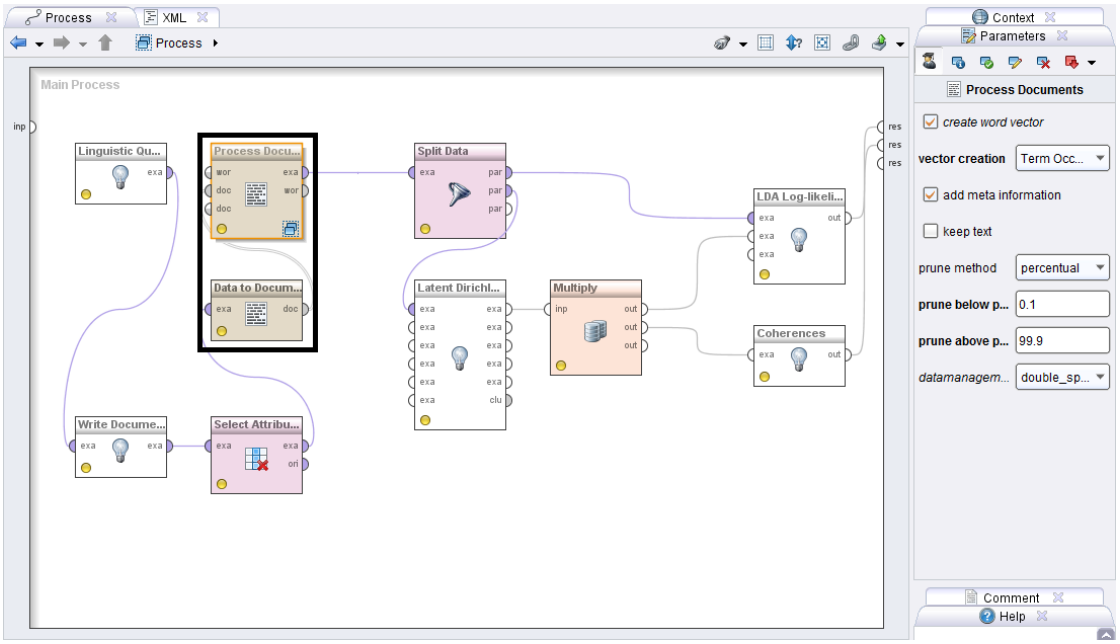


Figure 0.5: Generation of Word-Vectors from example set with text attribute.

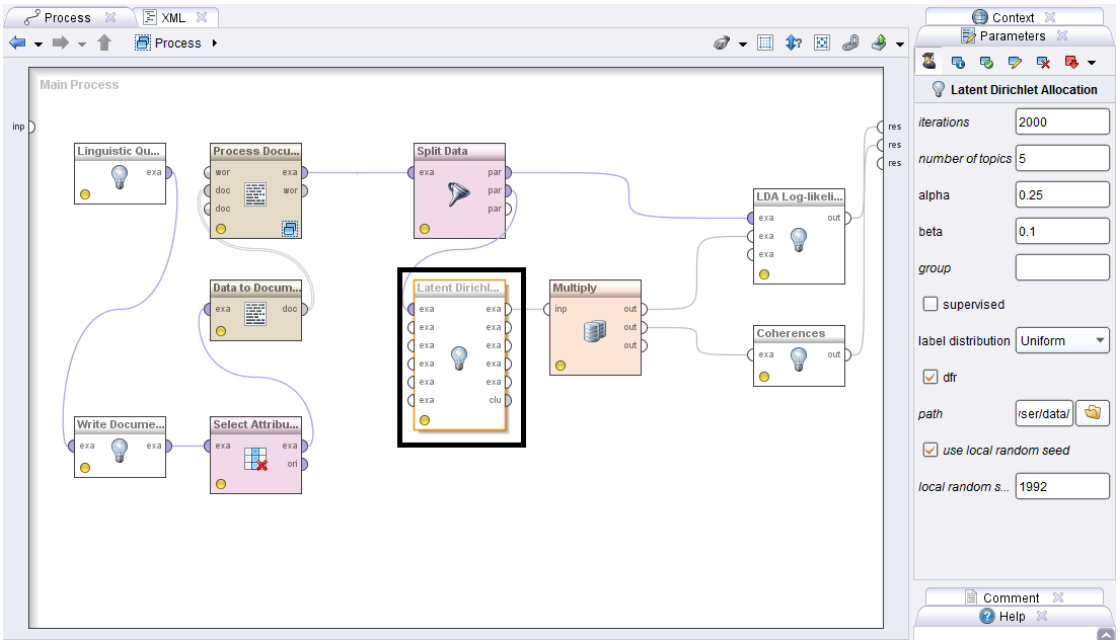


Figure 0.6: Latent Dirichlet Allocation operator to extract latent topics from a document collections given as Bag-of-Words.

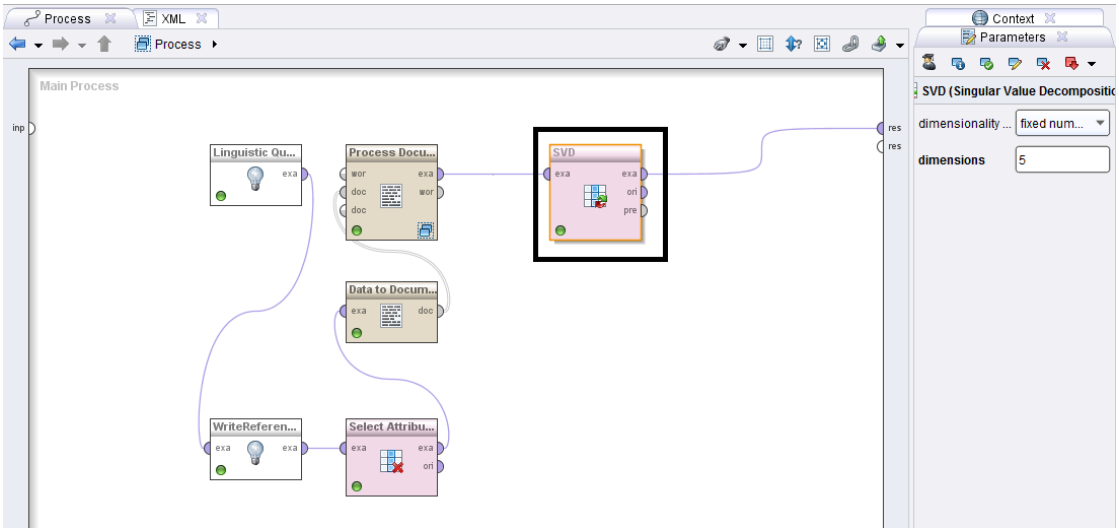


Figure 0.7: Extraction of latent factors via Singular Value Decomposition operator from RapidMiner.

topics by Gibbs sampling. Additional an operator that performs Variational Information for LDA is also available. Although we implemented both variants, in our experiments we used Gibbs sampling that showed good performance. For the Gibbs sampler, we need to specify how many iterations we want to process. Further, the number of topics to be extracted and the meta parameters of the Dirichlet priors need to by specified. For standard LDA, we un-check the supervised check box. For supervised LDA (sLDA) like temporal topic modelling, we check the supervised check box and specify the label distribution (Uniform, Beta, Gompertz). If we use sLDA, the input example set must contains a label attribute additional to the Word-Vectors. For visualization of the results we check the **dfr** check box and specify the path to the data folder for the DFR-Browser. Figure 0.6 shows the Latent Dirichlet Allocation Operator in a process. The resulting example sets of this operator contain the topic-distributions, the document-topic distributions and the estimated parameters for the label distribution for sLDA.

Latent Factor Models

Using the Word-Vectors collected into a Term-Document Matrix, we can easily perform LSA via a SVD. The RapidMiner operator **Singular Value Decomposition** extracts the singular values and the singular vectors from an example set. This example set is used as numeric matrix. The operator extracts only the left singular vectors. To extract the right singular vectors, we transpose the data set by the **Transpose** operator and apply the SVD. Given the number of components (factors to be extracted), the operator result is an example set containing the singular vectors and an example set containing the singular values. In Figure 0.14, we illustrate the operator in an example process.

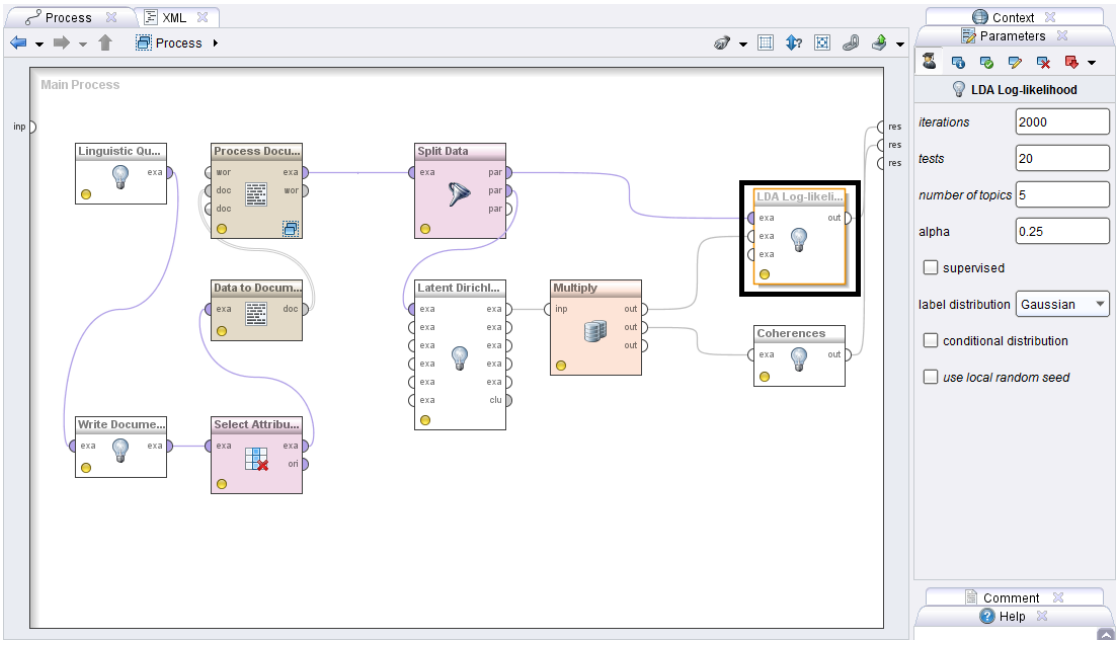


Figure 0.8: Log-likelihood operator to calculate the likelihood of a test document collections based on sequential Monte Carlo sampling.

Evaluation

Different methods to evaluate latent factor and latent topic models can be used via the **Coherence** operator and the **LDA Log-likelihood** operator, respectively the **LSA Log-likelihood** operator. The Coherence operator takes as input an example set containing word probabilities for topics or factors as a result from the Latent Dirichlet Allocation and the SVD operator. We leverage the Palmetto Toolbox [8] to estimate the different coherence measures based on the top words. From the word probabilities the most likeliest words (the number is given by a parameter) are used for the coherences. To use this operator we need a Lucence-based index from a large text collection that is used as reference. We use the Wikipedia articles to generate such an index that contains coherence values using co-occurrences and relative frequencies. We calculated such indices from German and English using the Palmetto library and the Wikipedia corpora from the Institute of the German Language. The LDA/LSA Log-likelihood Operator need no additional resources. We calculate the likelihoods of a test set of documents by Sequential Monte Carlo methods. As input, the LDA Log-likelihood Operator takes a set of test documents as Word-Vectors with occurrence data in an example set and the word-topic distributions resulting from the LDA operator. The number of iterations specifies the number of Monte Carlo Samples for the estimation of the likelihood. To reduce variance in the likelihood estimation, a number of independent tests are performed. The result is an example set that contains for each test the log-likelihood. The LSA Log-likelihood Operator takes as input an example set with Word-Vectors as test input and the factor

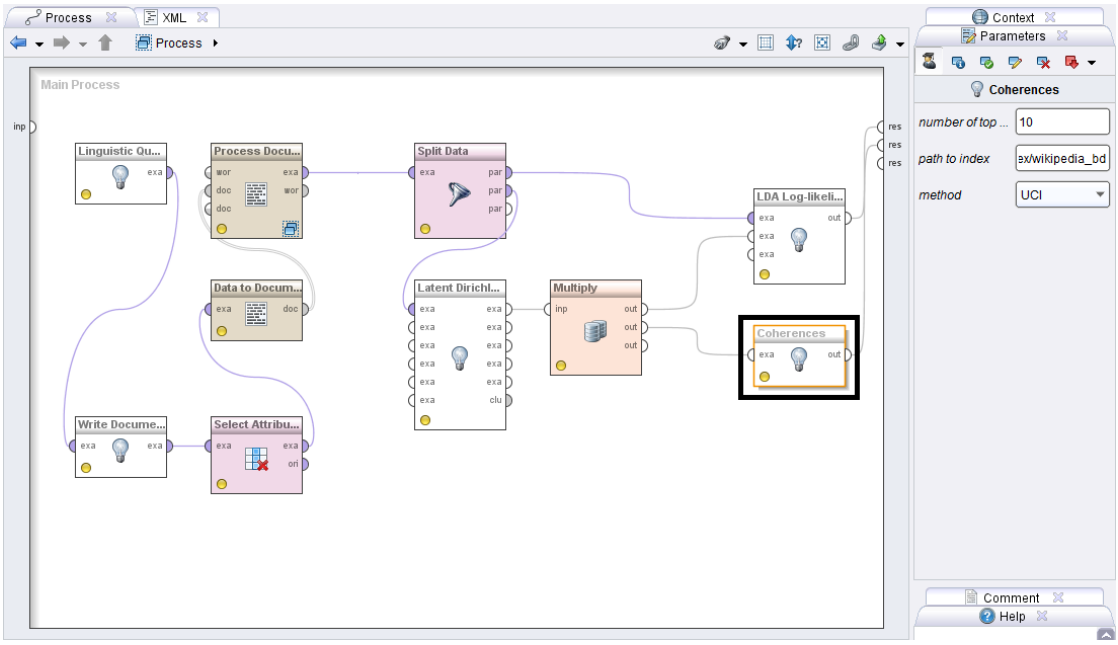


Figure 0.9: Coherence operator to estimate standard coherence measures using the Palmetto library.

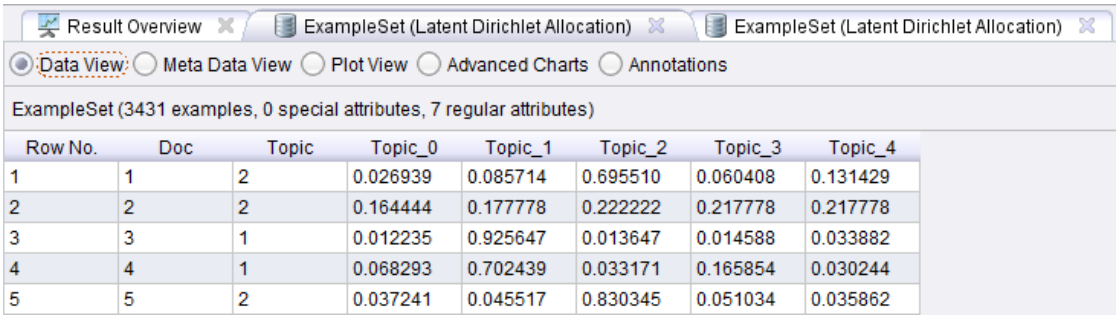
representation of the words as second input. This factor representation are for example the left-singular vectors from the Term-Document Matrix extracted by a Singular Value Decomposition. The LSA Log-likelihood Operator performs also Sequential Monte Carlo methods but the probabilities are based on distances in the factor representation of the documents and the words. As additional parameters, both operators can estimate joint likelihoods of the documents and possible given labels like time stamps.

Results

The results from the latent factor and latent topic models can be use either in tabular or example set form or in special formats for visualization. Using the results as it is given from the topic models operators, we get two example set containing the topic-word distributions and document-topic distribution. As additional attribute we report the most likeliest topic for each word and each document in the example sets. For factor models using for instance Singular Value Decomposition on the Term-Document Matrix, the factors are given as vectors in an example set and can be used in similar ways as the results from the topic models. In the Figures 0.10 and 0.11, the example sets from the results of LDA are shown as they are internally represented in RapidMiner.

Additional, we implemented an export of the results from the latent variable methods and the corpora for visualization by the DFR-Browser from Andrew Goldstone[2]. To process the documents from the corpus for information extraction needed for the

[2]https://github.com/agoldst/dfr-browser



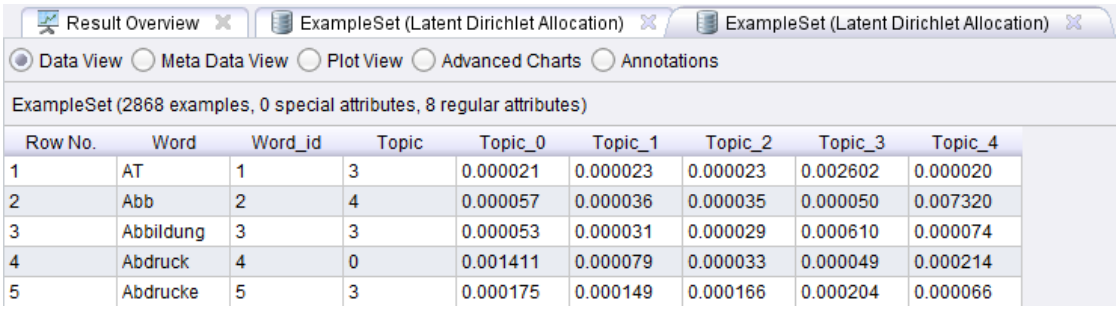
Result Overview | ExampleSet (Latent Dirichlet Allocation) | ExampleSet (Latent Dirichlet Allocation)

Data View | Meta Data View | Plot View | Advanced Charts | Annotations

ExampleSet (3431 examples, 0 special attributes, 7 regular attributes)

Row No.	Doc	Topic	Topic_0	Topic_1	Topic_2	Topic_3	Topic_4
1	1	2	0.026939	0.085714	0.695510	0.060408	0.131429
2	2	2	0.164444	0.177778	0.222222	0.217778	0.217778
3	3	1	0.012235	0.925647	0.013647	0.014588	0.033882
4	4	1	0.068293	0.702439	0.033171	0.165854	0.030244
5	5	2	0.037241	0.045517	0.830345	0.051034	0.035862

Figure 0.10: Document-topic distribution: For each document, we one examples contains the document number, the distribution over the topics and the most likeliest topic (Topic).



Result Overview | ExampleSet (Latent Dirichlet Allocation) | ExampleSet (Latent Dirichlet Allocation)

Data View | Meta Data View | Plot View | Advanced Charts | Annotations

ExampleSet (2868 examples, 0 special attributes, 8 regular attributes)

Row No.	Word	Word_id	Topic	Topic_0	Topic_1	Topic_2	Topic_3	Topic_4
1	AT	1	3	0.000021	0.000023	0.000023	0.002602	0.000020
2	Abb	2	4	0.000057	0.000036	0.000035	0.000050	0.007320
3	Abbildung	3	3	0.000053	0.000031	0.000029	0.000610	0.000074
4	Abdruck	4	0	0.001411	0.000079	0.000033	0.000049	0.000214
5	Abdrucke	5	3	0.000175	0.000149	0.000166	0.000204	0.000066

Figure 0.11: Topic-word distribution: For each word, we one examples contains the word, the word id, the distribution over the topics and the most likeliest topic (Topic).

visualization, we implemented the **Write Document Reference** operator. As shown in Figure 0.12, from an example set containing texts with information about a title, author, publication date and source as attribute information are saved locally where the visualization tool DFR-Browser finds them. The DFR-Browser can be started as a web server and the visualization can be seen in web browser like Firefox.

Diachronic Linguistic Process

To perform diachronic linguistic tasks, we use the Latent Dirichlet Allocation operator for sLDA. From the linguistic corpora, we take the information about publication date to extract labels for each document. The attribute **date** from the resulting example set from the TEI or Linguistic Query Operator contains the time information as string. First, we need to convert this into a numerical value by the operators **Nominal to Date** and **Date to Numeric**. Here, the concrete date format (for example "yyyy-MM-dd") must be given and the we need to specify the time unit into which we transform the date to numeric (for example years since 1900). This is illustrated on the left in Figure 0.13 (most important operators are framed). After the extraction of the information for

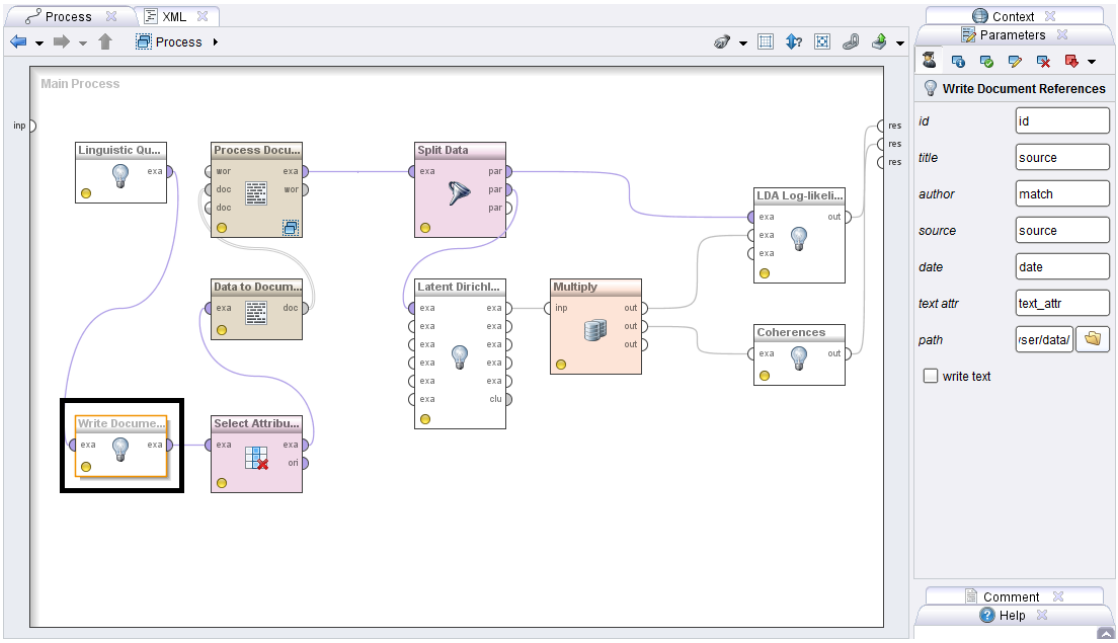


Figure 0.12: Write Document Reference operator: Writes formatted references from the document collection for visualization by the DFR-Browser.

visualization by the Write Document References operators, we select the text attribute **text_attr** and the date attribute by the **Select Attribute** operator and process the text to Word-Vectors by the text processing operators. Before the date attribute can be used for temporal topic modeling, we need to assign it to the **label** role via the **Set Role** operator. Now, the documents can be used together with the date attribute to extract topics and to estimate label distributions. For temporal topic modeling, we need to check the **supervised** check box in the Latent Dirichlet Allocation operator. We also need to specify with which distribution the labels shall be modeled. For time stamps we can use the Beta, the Uniform and the Gompertz distribution. As results, we get besides the topic-word distributions and the document-topic distributions also the parameters that are estimated by Maximum Likelihood Estimation (MLE) during the topic modeling for the corresponding label distribution.

Variety Linguistic Process

For variety linguistic tasks to compare and match text collections, we implemented factor models with distribution matching in the **Distribution Matching** operator. Given the Word-Vector representations of two text collection the operator extracts latent factors such that on the subspace spanned by these factors the documents from both collections have a similar distribution. The operator expects two inputs. The first input is a Term-Document Matrix as example set from a text collection with a certain distribution.

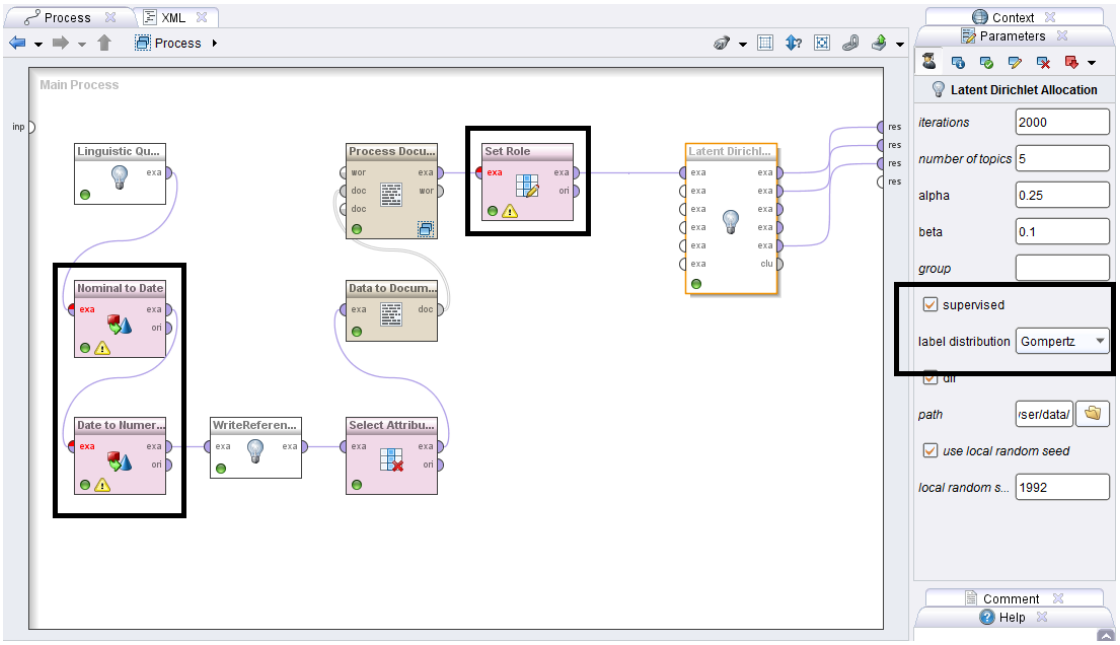


Figure 0.13: Process for Diachronic Linguistic: From a linguistic data source, we retrieve a KWIC-list with time information. The date is used as numeric label in temporal topic modelling.

The second input is a Term-Document Matrix from a second text collection with a different distribution. The results are two example sets containing the projections of the Word-Vectors from the document collections onto the subspace spanned by the factors. In Figure 0.14, an example process for variety linguistic by distribution matching is shown. There are two implementations available. First, a distribution match based on a Singular Value Decomposition extracts factors as the singular vectors of the union of both term-document matrices. Second, we implemented an online method for distribution matching, by efficiently solving an optimization problem through Stochastic Gradient Descent directly on a matrix manifold. We implemented the SGD in Matlab in the ManOpt library [3] for general Riemann manifolds. To use this method, we need Matlab to be installed and the ManOpt library.

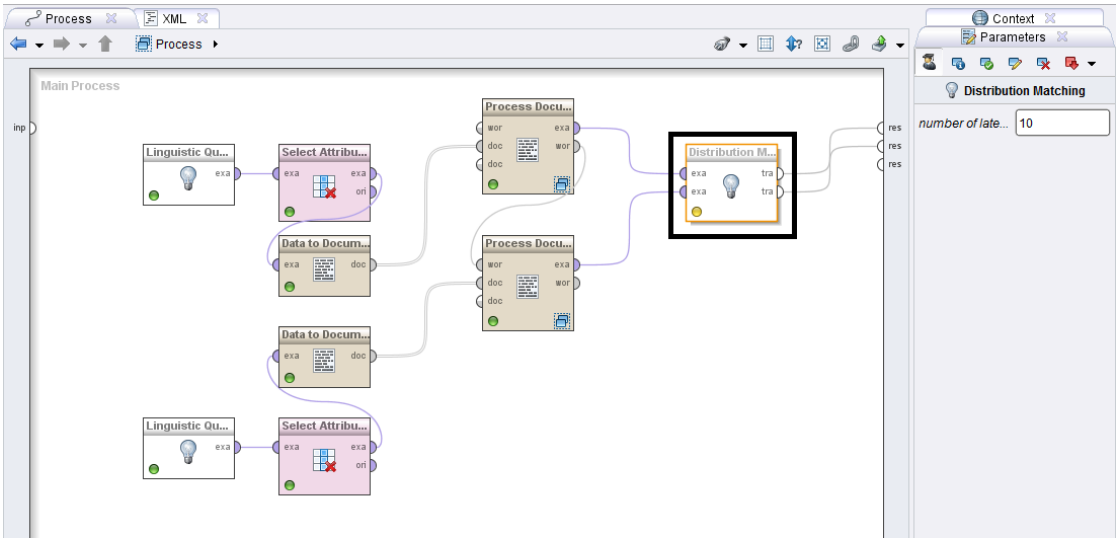


Figure 0.14: Process for Variety Linguistic.

Bibliography

[1] Michael Beißwenger, Maria Ermakova, Alexander Geyken, Lothar Lemnitzer, and Angelika Storrer. A tei schema for the representation of computer-mediated communication. *Journal of the Text Encoding Initiative*, (3), 2012.

[2] Michael Beiswenger, Harald Lüngen, Eliza Margaretha, and Christian Pölitz. Mining corpora of computer-mediated communication. In Gertrud Faaß and Josef Ruppenhofer, editors, *Proceedings of the 12th edition of the KONVENS conference Vol. 1*, Analysis of linguistic features in Wikipedia talk pages using machine learning methods, pages 42 – 47, 2014.

[3] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt: a Matlab toolbox for optimization on manifolds. *arXiv preprint arXiv:1308.5200 [cs.MS]*, 2013.

[4] Jörg Didakowski; Alexander Geyken. From dwds corpora to a german word profile methodological problems and solutions. In *In Network Strategies, Access Structures and Automatic Extraction of Lexicographical Information*, pages 43–52. Mannheim: Institut fr Deutsche Sprache. (OPAL - Online publizierte Arbeiten zur Linguistik X/2012), 2013.

[5] Markus Hofmann and Ralf Klinkenberg. *RapidMiner: Data Mining Use Cases and Business Analytics Applications*. Chapman & Hall/CRC, 2013.

[6] Eliza Margaretha and Harald Lüngen. Building Linguistic Corpora from Wikipedia Articles and Discussions. *JLCL*, 29(2):59–82, 2014.

[7] Matthew North. *Data mining for the masses*. 2012.

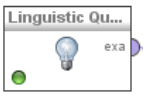
[8] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the eight International Conference on Web Search and Data Mining, Shanghai, February 2-6*, 2015.

[9] A Sokirko. Ddc - a search engine for linguistically annotated corpora. *Dialogue*, 2003.

0.1 Operator Reference

Next, we summarize the operators RapidMiner Corpus Linguistic Plugin. For clarity, we group the references into subsections.

0.1.1 Data Imports



Linguistic Query Operator : Interface to a linguistic corpora via a data base server as maintained by Berlin Brandenburg Academia of Science

Parameter	Description
query	The linguistic query on the corpora.
encapsulation	Character that indicates position of query match in the results.
sample size	Number of snippets in the results KWIC-list.
encoding	Character encoding (UTF-8)
data source	Corpus to query.
context size	Number of sentences before and after the sentence that contains the query match.
extract lemmas	Retrieve additional lemma information for each word (if available).
extract tags	Retrieve additional Parts-of-Speech for each word (if available).
Output	Example set containing KWIC-list.

TEI Query Operator: Stream reader for TEI formatted XML-files

Parameter	Description
query	The query as regular expression on the TEI-file.
file	Path to the TEI-file to be queried.
context	Environment in which we look for query match (posting, paragraph or sentence level).
context size	Number of characters before and after the regular expression match.
sample size	Number of snippets in the results KWIC-list.
regular expression	Query input mask regular expression on the TEI-file (with editor).
encoding	Character encoding (UTF-8)
output	Example set containing KWIC-list.

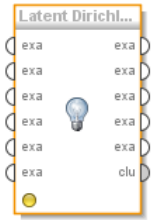
WordNet Operator: Query word relations from word net files

Parameter	Description
query	The word for the WordNet relations.
word net resource	Path to the word net data base files.
output	Example set containing word net relations.

Word Profiles Operator: Query word profiles from DWDS server

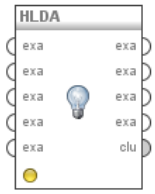
Parameter	Description
query	The word for the word profiles relations.
number of results	The number of related words by word profiles.
output	Example set containing word profiles relations.

0.1.2 Latent Topic Models



Latent Dirichlet Allocation Operator: Extracts latent topics via LDA.

Parameter	Description
iterations	The number of iterations for the Gibbs sampler.
number of topics	The number of latent topics to be extracted.
alpha	Meta parameter for Dirichlet prior on document-topic distribution.
beta	Meta parameter for Dirichlet prior of topic-word distribution.
group	Group attribute in data set.
supervised	Supervised LDA or unsupervised.
label distribution	The distribution of the document labels (Gauss, Beta, Uniform, Gompertz).
dfr	Print results in format for DFR-Browser.
path	Path to save files form DFR-Browser.
use local random seed	Use local random seed (for reproducibility).
local random seed	The concrete seed for the random number generator.
input 1	Example set of documents as word vectors with occurrences. (for supervised an additional attribute with label role must be available.)
output 1	Example set containing topic-word distributions.
output 2	Example set containing document-topic distributions.
output 3	Example set containing number of assignments of topics to words (for evaluation).
output 4	Example set containing number of assignments of topics to any word (for evaluation).
output 5	Example set containing parameters of the estimated label distributions.



Hierarchical Latent Dirichlet Allocation
Operator: Extracts latent topics via LDA including hierarchies between the topics.

Parameter	Description
iterations	The number of iterations for the Gibbs sampler.
number of topics	The number of latent topics to be extracted.
alpha	Meta parameter for Dirichlet prior on document-topic distribution.
beta	Meta parameter for Dirichlet prior of topic-word distribution.
group	Group attribute in data set.
supervised	Supervised LDA or unsupervised.
label distribution	The distribution of the document labels (Gauss, Beta, Uniform, Gompertz).
dfr	Print results in format for DFR-Browser.
path	Path to save files form DFR-Browser.
use local random seed	Use local random seed (for reproducibility).
local random seed	The concrete seed for the random number generator.
input 1	Example set of documents as word vectors with occurrences. (for supervised an additional attribute with label role must be available.)
output 1	Example set containing topic-word distributions.
output 2	Example set containing document-topic distributions.
output 3	Example set containing number of assignments of topics to words (for evaluation).
output 4	Example set containing number of assignments of topics to any word (for evaluation).
output 5	Example set containing parameters of the estimated label distributions.

Latent Dirichlet Allocation with Word Features Operator: Extracts latent topics via LDA and includes word features and relations via priors.

Parameter	**Description**
iterations	The number of iterations for the Gibbs sampler.
number of topics	The number of latent topics to be extracted.
alpha	Meta parameter for Dirichlet prior on document-topic distribution.
beta	Meta parameter for Dirichlet prior of topic-word distribution.
lambda	
gamma	
number of word groups	for group lasso based prior.
a	Meta parameter for group lasso penalty.
prior	used prior
dfr	Print results in format for DFR-Browser.
path	Path to save files form DFR-Browser.
use local random seed	Use local random seed (for reproducibility).
local random seed	The concrete seed for the random number generator.
input 1	Example set of documents as word vectors with occurrences. (for supervised an additional attribute with label role must be available.)
input 2	Example set containing word relations.
output 1	Example set containing topic-word distributions.
output 2	Example set containing document-topic distributions.
output 3	Example set containing number of assignments of topics to words (for evaluation).
output 4	Example set containing number of assignments of topics to any word (for evaluation).
output 5	Example set containing parameters of the estimated label distributions.

Dirichlet Multinomial Regression
Operator: Extracts latent topics via LDA.

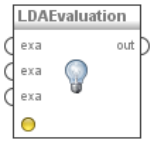
Parameter	Description
iterations	The number of iterations for the Gibbs sampler.
number of topics	The number of latent topics to be extracted.
alpha	Meta parameter for Dirichlet prior on document-topic distribution.
beta	Meta parameter for Dirichlet prior of topic-word distribution.
lambda	
sigma	
group	Group attribute in data set.
dfr	Print results in format for DFR-Browser.
path	Path to save files form DFR-Browser.
use local random seed	Use local random seed (for reproducibility).
local random seed	The concrete seed for the random number generator.
input 1	Example set of documents as word vectors with occurrences.
input 2	Additional document attributes.
output 1	Example set containing topic-word distributions.
output 2	Example set containing document-topic distributions.
output 3	Example set containing number of assignments of topics to words (for evaluation).
output 4	Example set containing number of assignments of topics to any word (for evaluation).
output 5	Example set containing parameters of the estimated label distributions.

0.1.3 Latent Factor Models

Distribution Matching: Extracts latent factors that match distributions.

Parameter	Description
number of factors	The number latent factors to be extracted.
input 1	Example set of documents as word vectors from a certain distribution.
input 2	Example set of documents as word vectors from another distribution.
output 1	Example set of the documents from input 1 projected into the latent factor presentation.
output 2	Example set of the documents from input 2 projected into the latent factor presentation.

0.1.4 Evaluation Methods



LDA Log-likelihood Operator: Estimates log-likelihood on a test collection of sequences of words in document given the results of LDA.

Parameter	Description
iterations	The number of iterations for the Gibbs sampler.
tests	The number of random tests.
number of topics	The number to be extracted.
alpha	Meta parameter for Dirichlet prior on document-topic distribution.
supervised	Supervised LDA or unsupervised.
label distribution	The distribution of the document labels (Gauss, Beta, Uniform, Gompertz).
conditional distribution	
use local random seed	Use local random seed (for reproducibility).
local random seed	The concrete seed for the random number generator.
input 1	Example set of test documents as BoW with occurrence data
input 2	Example set of word-topic distribution from an LDA result.
output 1	Example set containing log-likelihoods from each test.

LSA Log-likelihood Operator: Estimates log-likelihood on a test collection of word-vectors by distance based distribution estimation.

Parameter	Description
iterations	The number of iterations for the Gibbs sampler.
tests	The number of random tests.
number of topics	The number to be extracted.
alpha	Meta parameter for Dirichlet prior on document-topic distribution.
smoothing	
gamma	
supervised	Supervised LSA (PLS) or unsupervised.
use local random seed	Use local random seed (for reproducibility).
local random seed	The concrete seed for the random number generator.
input 1	Example set of test word-vectors.
input 2	Example set of words in factor representation.
output 1	Example set containing log-likelihoods from each test.

Coherence Operator: Estimates standard coherence values based on top ranked word in each topic.

Parameter	Description
number of top words	The number of top ranked words in each topic used to estimate coherence values.
path to index	The path to the lucene index files for Palmetto.
method	Used coherence measure. (UCI,UMass,NPMI)
alpha	Meta parameter for Dirichlet prior on document-topic distribution.
input 1	Example set of topic-word distributions from an LDA results.
output 1	Example set containing the coherence value for each topic.