Analysis of academic literature on parsing by means of topic analysis

Diplomarbeit

zur Erlangung des Grades eines Diplom-Informatikers
im Studiengang Wirtschaftsinformatik

vorgelegt von

Thomas Schnitzler

Erstgutachter:	Prof. Dr. Ralf Lämmel Institut für Informatik
Zweitgutachter:	M. Sc. Johannes Härtel Institut für Informatik

Koblenz, im September 2017

Erklärung

Hiermit bestätige ich, dass die vorliegende Arbeit von mir selbständig verfasst wurde und ich keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet–Quellen – benutzt habe und die Arbeit von mir vorher nicht in einem anderen Prüfungsverfahren eingereicht wurde. Die eingereichte schriftliche Fassung entspricht der auf dem elektronischen Speichermedium (DVD-ROM).

	Ja	Nein
Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.	☐	☐
Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.	☐	☐

.....

(Ort, Datum) (Thomas Schnitzler)

Abstract

This diploma thesis' goal is to implement a software tool that executes a topic analysis as follows: Collect, preprocess and analyze scientific articles semiautomatically in order to study and summarize a given category of a scientific field automatically.
The data collection is based on the theory of systematic mapping study, on methods of data mining in terms of preprocessing and on latent dirichlet allocation in terms of data analysis. The output of the analysis is converted to HTML in order to link the output also from webpages that are not part of the above mentioned software tool.
Topic analysis with the help of this tool is demonstrated based on a running example for Parsing.

Zusammenfassung

Das Thema dieser Diplomarbeit ist die Implementierung eines Softwarewerkzeugs zur Durchführung einer Themenanalyse (zu Englisch: topic analysis). Die Themenanalyse soll leisten, dass halbautomatisch eine Sammlung, eine Analysevorbereitung und eine Analyse von wissenschaftlichen Artikeln durchgeführt wird, um eine gegebene Kategorie eines wissenschaftlichen Gebietes automatisiert zu studieren und zusammenzufassen.
Die Sammlung der Daten beruht auf der Theorie der systematischen Abbildungsstudie (zu Englisch: systematic mapping study).
Die Analysevorbereitung auf Methoden des Data Minings und die Analyse selbst beruht auf latenter Dirichletallokation (zu Englisch: latent dirichlet allocation).
Die Ausgabe der Analyse wird umgewandelt in HTML, um von Webseiten außerhalb des Softwarewerkzeugs einsehbar zu sein.
Die Durchführung einer Themenanalyse mithilfe des oben angegebenen Softwarewerkzeugs wird mithilfe eines Laufbeispiels für Parsing demonstriert.

Acknowledgement

I would like to thank Mr. Prof. Dr. Ralf Lämmel for this very interesting topic for my diploma thesis. I would also like to thank Mr. Prof. Dr. Ralf Lämmel, Mrs. Prof. Dr. Maria A. Wimmer, Mr. Prof. Dr. Klaus Diller and Mr. Prof. Dr. Stefan Müller who were a big encouragement to me to finish my studies. I also thank Mr. M.Sc. Johannes Härtel who was very attentive with the intention to provide me with the right information to adapt this diploma thesis to the conceptual needs of a diploma thesis for the software language engineering working group of the University Koblenz-Landau.

I thank my parents, my brother, my grandparents and anyone who helped me during my studies.

For Michaela

Contents

Chapter 1

Motivation and Introduction

Getting to know a scientific field is a longterm task because the majority of scientific fields have a long history. If a researcher had the task to summarize the contents of compiler building then the researcher could use each category of compiler building which e.g. is Parsing in order to study and summarize each category of compiler building based on scientific sources like scientific articles.
In order to assist this study and summary of a category of a scientific field it is desired to implement a semiautomatic tool that

- collects attributes of scientific articles like "authors", "title", "year of publication", "conference", abstracttext and fulltext,
- preprocesses abstracttexts or, if available, fulltexts in order to
- analyze the preprocessed abstracttexts or, if available, fulltexts

according to the data mining solution in figure 1.1 which is copied from [1, page 4].

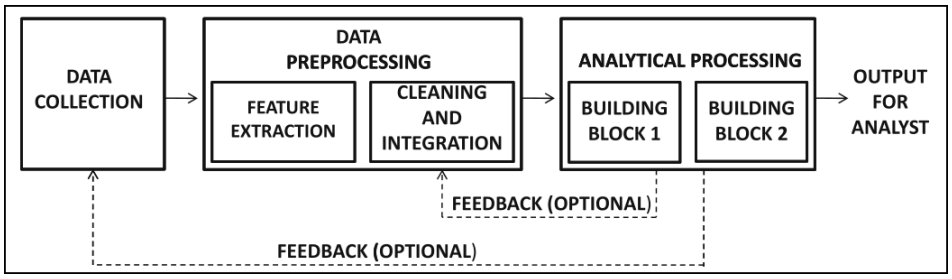


Figure 1.1 The data processing pipeline which is a copy from [1, page 4]

A semiautomatic tool which is an assistant for a researcher while studying and summarizing categories of scientific fields is not only related to data mining procedures as illustrated in figure 1.1. This semiautomatic tool is an implementation of topic analysis which is introduced in section 2.3. Topic analysis is a correlation of systematic mapping study presented in section 2.1 with latent dirichlet allocation introduced in section 2.2. The procedure of the systematic mapping study as it is illustrated in figure 1.2 and copied from [5, slide 11] helps to collect correct values of scientific articles consisting of the attributes "authors", "title", "year of publication", "conference", "abstracttext" and "fulltext" with the help of research questions and search strings.

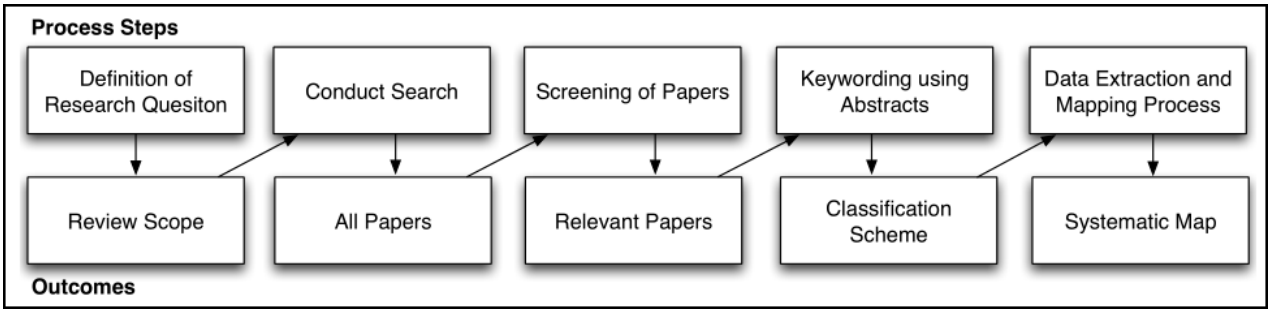


Figure 1.2 The sequence of the systematic mapping study which is a copy from [5, slide 11]

If any value is collected at the above listed attributes then abstracttexts and fulltexts will be preprocessed and analyzed as illustrated in figure 1.1. In line of the topic analysis the analysis is executed by latent dirichlet allocation. An output of a finished latent dirichlet allocation procedure is also created and transferred to HTML in order to link the output to other webpages. An example for this output is illustrated in figure 1.3 and in figure 1.4 below.
Our intention is to implement a semiautomatic tool that is able to do topic analysis. The tool's

- fundamentals are introduced in section 2 concerning the background,
- implementation is presented in section 3,
- demonstration for "Parsing" as a running example is presented in section 4 and

- evaluation in terms of a conclusion is summarized in section 5.

We also give some notice for future works concerning the topic analysis tool in section 5.2.

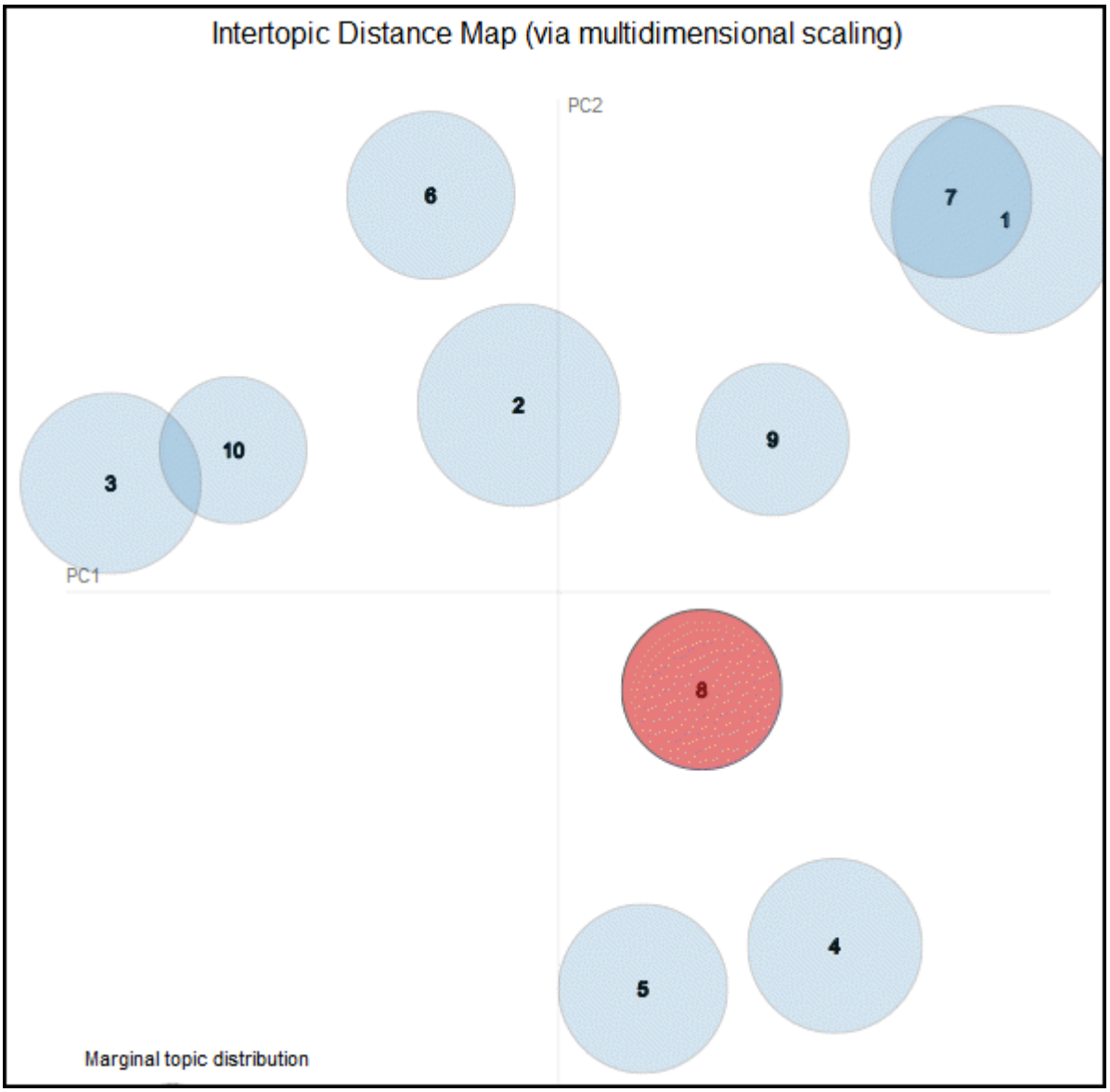


Figure 1.3 10 Topics for "Parsing" with the help of latent dirichlet allocation

What is nwe?

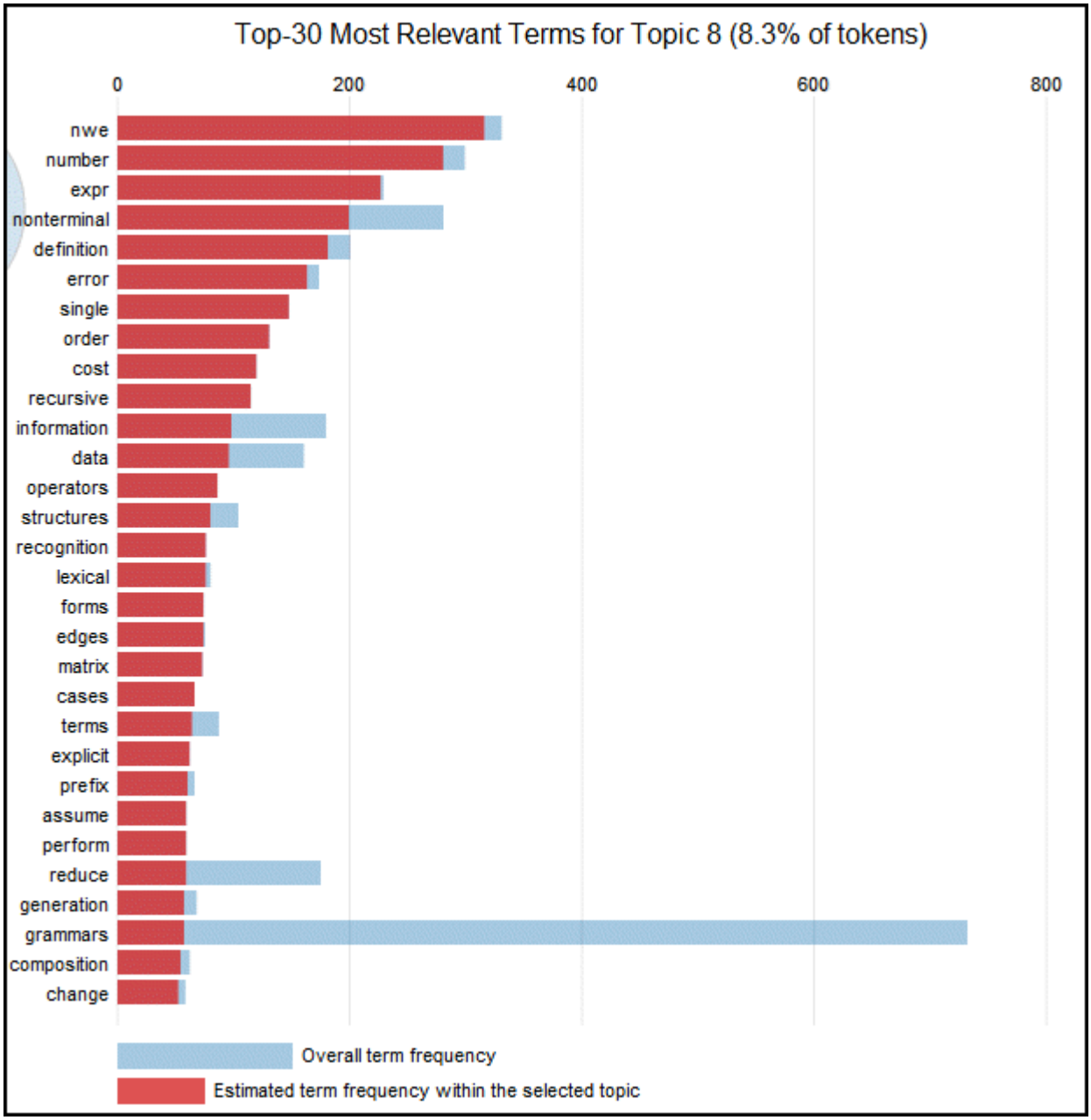


Figure 1.4 Topic 8 of "Parsing" with the help of R, [12], [4], [7] and [6]

Chapter 2

Background

The topic analysis tool introduced in section 1 is based on a systematic mapping study which is introduced in section 2.1. The analysis part of the systematic mapping study is provided by latent dirichlet allocation. A basic introduction of latent dirichlet allocation which is useful for understanding the analysis part of the topic analysis tool is presented in section 2.2.
The topic analysis tool executes topic analysis which is a correlation of systematic mapping study with latent dirichlet allocation. A description of topic analysis is presented in section 2.3.

2.1 Introduction to systematic mapping study

The systematic mapping study is the methodology for the topic analysis tool introduced in section 1 and it represents the function to study a category of a scientific field in order to "get an overview of a certain research area and how far it's covered in research" as it is quoted from [5, slide 3]. The systematic mapping study's approach for this aim is to "study the research field by using methods from information retrieval and statistical analysis" which is a quotation from [5, slide 3]. Statistical analysis is necessary for the category's summary. The information retrieval methods are the methods to find the right sources in order to study and summarize them subsequently. Our information retrieval methods for finding the right sources to study and summarize are derived from the systematic mapping study's sequence illustrated in figure 2.1.

Some general idea is missing here as to how SMS and topic analysis are really to be married?

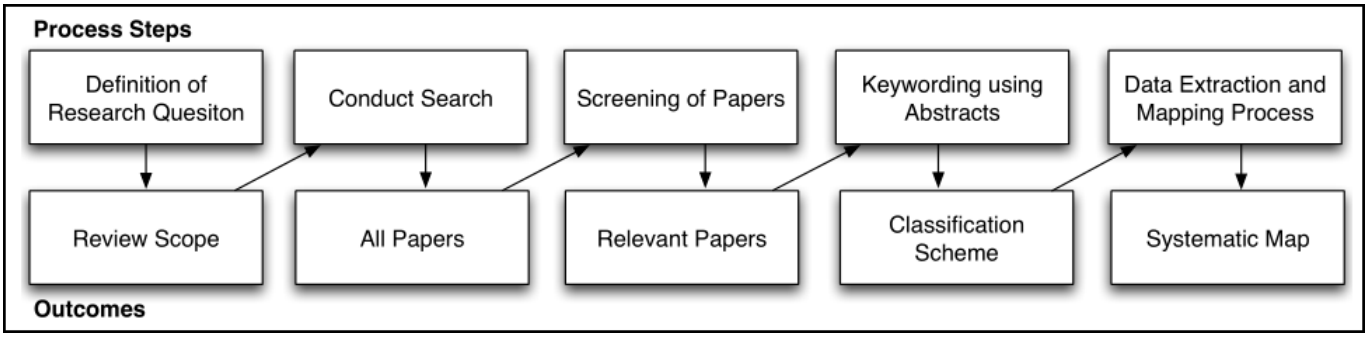


Figure 2.1 The systematic map which is copied from [5, slide 11]

Again?

First of all the quality standards for sources to study and the potential provider for the sources to study have to be determined (see [5, slide 13-16]). The output of this quality assurance is the *review scope* which will be the selection of sources with limited possibilities according to the above mentioned qualitiy standards if the search is conducted. This implies that the conducted search is guided by the review scope from the systematic mapping study's quality assurance from above. The conducted search as the second systematic mapping study's procedure step is based on information retrieval methods, i.e. web search engines have to be used in order to find the sources to study, or in other words search strings, have to be formulated for and submitted to scientific databases in order that the scientific articles'

- authors,
- titles,
- year of publication,
- conference,
- abstract or
- fulltext

Again?

are returned by these scientific databases.
The third systematic mapping study's step is to exclude those returned scientific articles from web search engines that are not compliant with the research

questions' limitations with regard to the systematic mapping study's quality assurance. The systematic mapping study's concept for exclusion is to systematically examine the search strings and search results after web search engines return search results. The examination of search results in terms of the exclusion is restricted to each scientific article's abstract or fulltext. After excluding, systematic mapping study generates keywords for each scientific article that is used to generate classes in terms of studying. Then it is summarized what is studied before. The summary closes that given keywords "keyword a" and "keyword b" from the scientific article named "article" form a class "class c" or two different classes "class c" and "class d".
The systematic map is the relation between keywords of classes (topics) and words in source documents that are equal to the keywords (compare with [5, slide 30]).

2.2 Latent dirichlet allocation

Latent dirichlet allocation is originally from [3] and is a possibility to let the computer study and summarize the topics of a text corpus without any human interaction, e.g. with the help of [6]. In order to achieve output that defines the scope of a category of a scientific field as it is wanted in terms of this thesis the input text corpus for latent dirichlet allocation must be from one category otherwise it is not clear what distinct category consists of a particular output topic of latent dirichlet allocation.
To get more technical information on latent dirichlet allocation the description of Microsoft's API for latent dirichlet allocation is copied from [9] to appendix section B.1.
[8, page 13] gives a short introduction to latent dirichlet allocation which is quoted here as follows: (-Beginning of quotation-) "LDA is a generative model, which means that it attempts to describe how a document is created. It is a probabilistic model because it says that a document is created by selecting topics and words according to probabilistic representations of natural text. For instance, the words that I use to write this paragraph pertain to a subtopic of this entire paper as a whole. The actual words I use to compose it are chosen based on that topic. The inherent probability in modeling the selection of each word stems from the fact that natural language allows us to use multiple different words to express

the same idea. Expressing this idea under the LDA model, to create a document within a corpus, imagine that there is a distribution of topics. For each word in the document that is being generated, a topic is chosen from a Dirichlet distribution of topics. From that topic, a word is randomly chosen based on another probability distribution conditioned on that topic. This is repeated until the document is generated." (-End of quotation-)

2.3 Introduction to topic analysis

Topic analysis in terms of studying and summarizing a category of a scientific field as it is explained in section 2.1 consists of two tasks: Collecting data and analyzing data. The analysis of data relies on the data retrieval. The summary of the category as explained in section 2.1 relies on the analysis. The analysis is executed by latent dirichlet allocation which is introduced in section 2.2. Fetching the data for collection is executed by means of information retrieval where information retrieval in wikipedia's words at [14] "is the science of searching for information in a document, searching for documents themselves, and also searching for metadata that describe data, and for databases of texts, images or sounds". Generally we want to search for information in a document and search for documents as it is reflected in wikipedia's words above. However, the topic analysis tool as introduced in section 1 is more than information retrieval. It must implement the systematic mapping study's way to do information retrieval because as we learned in section 2.1 we must have quality assurance from the beginning until the end of systematic mapping study. This implies we assure that the fetched search results from web search engines do not collide with our research questions that we pose at the beginning of the systematic mapping study.
Moreover, we need a topic analysis environment for a scientific fields' category. Ordinary manual information retrieval does not have this quality assurance concept with the help of research questions and topic analysis environment before the search string is sent to the web search engine because the user enters a search string while thinking of how to formulate it.
In terms of systematic mapping study after we have the research questions for our topic analysis environment we formulate search strings in order to submit the search to a web search engine that is included in the search string. Any search string must be formulated advisedly because any web search engine's search re-

sults to each search string must be an answer to the formulated research questions (see [5, slide 21]). We take care that the scientific articles' attributes are filled with scientific articles' values in terms of fill and complete. If a search result does not answer a research question then this search result has to be excluded from the topic analysis environment. This will e.g. be the case if for the same search result an abstracttext as well as a fulltext are not available.
After the exclusion procedure the preprocessing in terms of topic analysis for the forthcoming analysis takes place. The goal of this preprocessing is to support the execution of latent dirichlet allocation which is the analysis of topic analysis. Preprocessing reads the raw abstracttexts and fulltexts from web search engines as input and writes as output the input which only consists of the text character

- between "a" and "z",
- between "A" and "Z",
- "space",
- "linefeed",
- "carriage return" and
- "fullstop".

A manual optimization for the preprocessed abstracttexts and fulltexts is also possible in terms of topic analysis: The user can modify and save the texts of abstracttexts or fulltexts as inputs for the analysis based on latent dirichlet allocation.
If preprocessing and optimization are finished for abstracttexts anf fulltexts of a topic analysis environment then a environment for latent dirichlet allocation can be added for this topic analysis environment. A latent dirichlet allocation environment can be taken for analysis, i.e. it selects the preprocessed abstracttexts and fulltexts of a topic analysis environment, executes latent dirichlet allocation in order to create output that can be linked to external webpages that are not part of the topic analysis tool and displayed.

Chapter 3

Main part

Based on the topic analysis in section section 2.3 we document the implementation of a topic analysis tool in this section. The requirements, the design and the implementation of the topic analysis tool are presented in order to understand what functionalities must be implemented in a tool in terms of the topic analysis introduced in section 2.3. The goal of implementing the topic analysis tool is to fetch scientific knowledge sources from the internet in order to study and summarize them automatically. The summary's result is a graphical output of the computations of latent dirichlet allocation in line with this topic analysis tool.

3.1 Requirements

In order to be able to implement the topic analysis tool with the functionalities illustrated in figure 3.1 the requirements for functionalities for the topic analysis tool must be as follows:

1. It must be possible to add any category of any scientific field as a topic analysis environment to the tool. The name for this topic analysis environment must be unique. There also must exist a possibility to provide research questions according to [5, slide 12-16]. According to [5, slide 17-19] search strings must be entered in an input form for the selected topic analysis environment. The search strings must be used to retrieve search results from web search engines. These search results must be transformed

to scientific articles' values for at least one field of the following scientific articles' attributes:

- "authors",
- "title" and
- "abstracttext"[1].

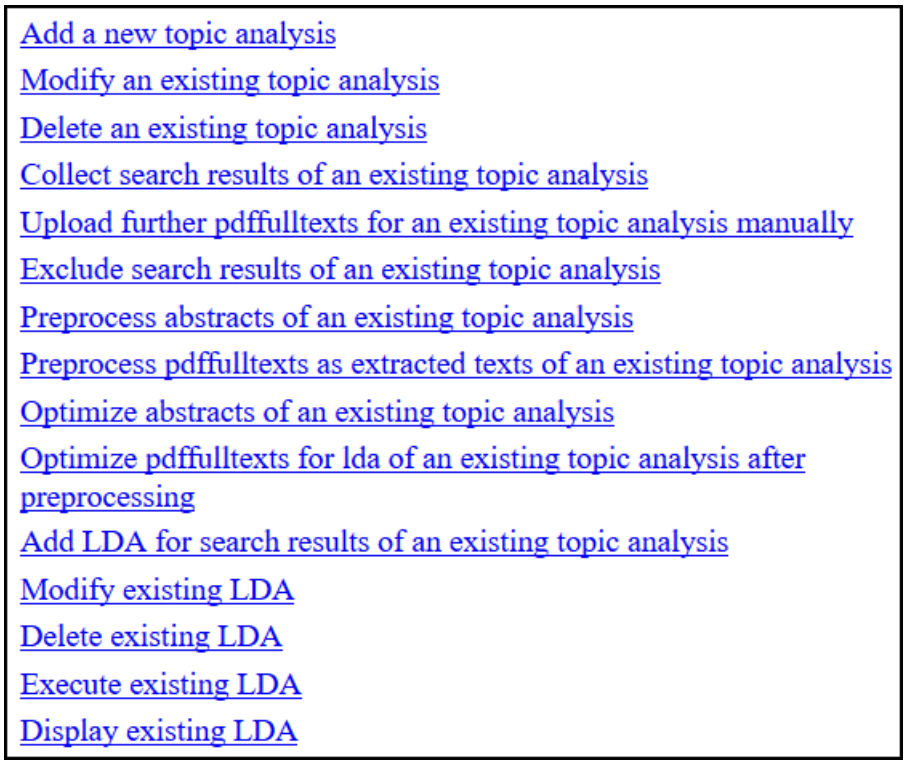


Figure 3.1 Functionalities for the topic analysis tool

2. Modifying a topic analysis environment must provide the possibility to select a topic analysis environment for a scientific field's category in order to modify

   - the name of the selected topic analysis environment,
   - research questions according to [5, slide 12-16] and

[1] This is what we mean with "collecting data" or "collecting values" in section 1 and in section 2.3.

- search strings according to [5, slide 17-19].

3. An added and/or modified topic analysis environment must be deletable which must affect any information that is associated with this topic analysis environment.

4. It must be possible to collect search results for an existing topic analysis environment. The prerequiste for the collection of search results is a selected topic analysis environment. After this selection it must be checked whether research questions and search strings exist for the selected topic analysis environment. After having checked any above listed prerequisite the following results must be reached:

   (a) Any search string that is provided in line of adding or modifying a topic analysis environment must be used for collecting search results at the topic analysis tool.

   (b) The search results must be saved into files that can be found by the topic analysis tool.

   (c) Each search result file must be transformed to search results for the topic analysis tool that can be used for further computations in the topic analysis tool.

5. It is desired to overcome a paywall of a web search engine legally which also is the web search engine for the collection of search results for the fulltext.

6. It must be possible for a user to upload additional fulltexts to existing search results if these fulltexts are the results of overcoming a paywall, e.g. with the help of a web search engine login as a member of the university Koblenz-Landau.

7. If an existing topic analysis environment is selected then the user must be able to exclude these search results that are not an answer to research questions entered for an added or modified topic analysis.

8. Existing abstracts and fulltexts of a topic analysis environment must be preprocessed in order to achieve the output which contains the following text character set:

- a-z,
- A-Z,
- fullpoint,
- space,
- linefeed or
- carriage return.

9. The user must have the possibility to optimize each text of any abstracttext and fulltext for a selected topic analysis environment. Optimization means that the user gets full access to each abstracttext and fulltext of a selected topic analysis environment with the purpose to delete any content of each abstracttext or fulltext that is not relevant for the analysis with the help of latent dirichlet allocation.

10. It must be possible to add an environment for latent dirichlet allocation. This environment for latent dirichlet allocation must contain the following parameters with the purpose to select abstracttexts and fulltexts for the computations of latent dirichlet allocacation:

    - "Conference" which is the reason for writing selected scientific articles and
    - period of time for the year of publication.

    Beside these selection parameters the name of the environment for latent dirichlet allocation and the amount of topics that latent dirichlet allocation must generate for a topic analysis environment must be saved.

11. The user must be able to modify the values for an existing environment for latent dirichlet allocation. Beside adding and modifying an environment for latent dirichlet allocation must be deletable.

12. The user must be able to execute latent dirichlet allocation based on the parameters saved in line of the existing environment for latent dirichlet allocation for saved, not excluded and preprocessed fulltexts or abstracttexts if particular fulltexts do not exist.

13. It must be possible to link the output of an executed latent dirichlet allocation to any possible webpage in the internet.

14. Keywords that are displayed as parts of topics according to point 13 must be related to chapters of fulltexts if the keywords are part of these chapters.

15. For the category "Parsing" a running example must be saved in the tool to be implemented. The running example has the following requirements:

    (a) The research question for "Parsing" must be as follows: "What key concepts will 'Parsing' contain if the scientific articles as part of the conferences

    - 'CC' = 'Compiler Construction',
    - 'SLE' = 'Software Language Engineering',
    - 'SCAM' = 'Source Code Analysis and Manipulation',
    - 'POPL' = 'Symposium on Principles of Programming Languages',
    - 'PLDI' = 'Conference on Programming Language Design and Implementation' and
    - 'ICFP' = 'International Conference on Functional Programming'

    are taken into consideration?"

    (b) Search results for studying and summarizing "Parsing" must be collected from the search engine `http://www.dblp.org/`.

3.2 Design

The design of the topic analysis tool is based on the software architecture in section 3.2.1, the database structure in section 3.2.2 and dependencies for functionalities in section 3.2.3 that are based on the requirements for functionalities for the topic analysis tool in section 3.1.

3.2.1 Choice of architecture

The software architecture of the topic analysis tool is a client-server-architecture based on Apache, PHP and MySQL. This architecture' advantage is that the communication between user und webserver is quick and easy because the user just

needs a webbrowser, a connection to the intranet or internet.
Forms for user inputs are very user-friendly.

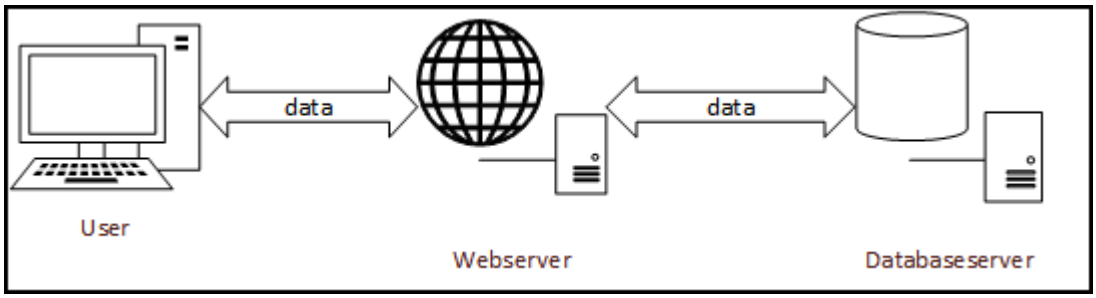


Figure 3.2 Client-server architecture for the topic analysis tool

Moreover, these forms are very easy to create and very easy to administrate.
The web- and databaseserver allow concurrent accesses to each service.
The communication between the topic analysis tool and the web search engines is not difficult to handle because the topic analysis tool is permanently connected to the internet.

3.2.2 Database structure

The entity-relationship model of the database backend of the topic analysis tool is as it is illustrated in figure 3.3[2]. In the following the purpose of each table based on figure 3.3 is described. The name of the topic analysis environment must be inserted in the table "topic_analysis" if a new topic analysis environment is added. The research questions and the search strings for a new or an existing topic analysis environment are saved as follows:

- Research questions are inserted in table "research_questions" which has n identifiers related to 1 identifier in table "topic_analysis".
- Search strings for filling the topic analysis tool with search results from a web search engine in terms of fill and complete are inserted in table "search_strings" which has n identifiers related to 1 identifier in table "topic_analysis".

[2]The E-R-model of this database is created with SQL Power Architect 1.0.7.

- Search strings for completing the topic analysis tool with search results from a web search engine are inserted in table "search_strings_for_results" which has n identifiers related to 1 identifier in table "topic_analysis".

Table "search_results" is the central table for collecting search results from web search engines. One row in table "search_results" is one scientific article. One column is one attribute of any scientific article saved in table "search_results". The table "search_results" has the scientific articles' attributes "authors", "title", "year of publication", "conference", "abstracttext" and "fulltext". The table "search_results" is the only table which is accessed by the environment for latent dirichlet allocation because it copies the content of "fulltext" to the input file folder for latent dirichlet allocation if fulltext exist for a given selection otherwise the content of "abstracttext" is copied.
The table "search_results" has n identifier related to 1 identifier in table "search_strings" in order to be able to reconstruct the origin of each search result. Moreover, the search results are related to table "topic_analysis" because table "search_results" has n identifier related to 1 identifier in table "search_strings". Table "search_strings" has n identifier related to 1 identifier in table "topic_analysis". For that reason any search result can be selected for a topic analysis environment if the identifier or the name of the topic analysis environment is given.
The parameters for latent dirichlet allocation are saved in table "lda". One environment for latent dirichlet allocation is one set of parameters with a unique name for this latent dirichlet allocation environment related to a topic analysis environment. 1 topic analysis environment can have n latent dirichlet allocation environments.
The computed selections based on the parameters in table "lda" are saved in table "search_strings_for_results" which are the affected scientific articles.
Information about relations in the database of the topic analysis tool are also available in the appendix section C.1[3].

[3]This database information is created with SQL Power Architect 1.0.7.

3.2.3 Dependencies for functionalities of the topic analysis tool

The dependencies for functionalities of the topic analysis are illustrated in figure 3.4. Referring to this illustration it must be noticed that

- the computation of each functionality takes place between two states,
- any functionality works for any possible category of any scientific field,
- computed values of a functionality are either stored in the database introduced in section 3.2.2 or in files where it can be found by a functionality that reads the values and
- the internal matters of each functionality are presented in section 3.3.

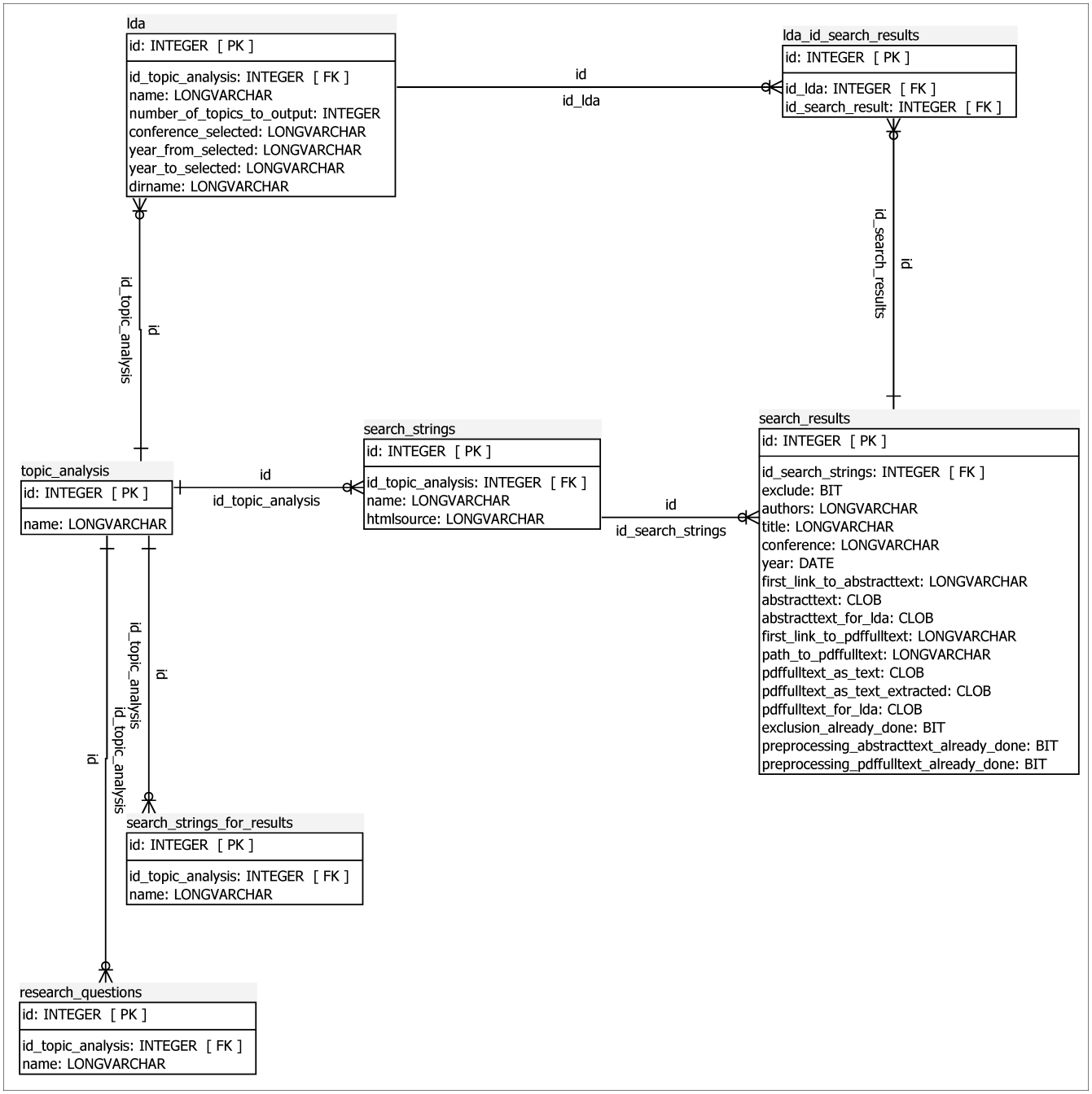


Figure 3.3 Entity-relational model of the database of the topic analysis tool created with SQL Power Architect 1.0.7

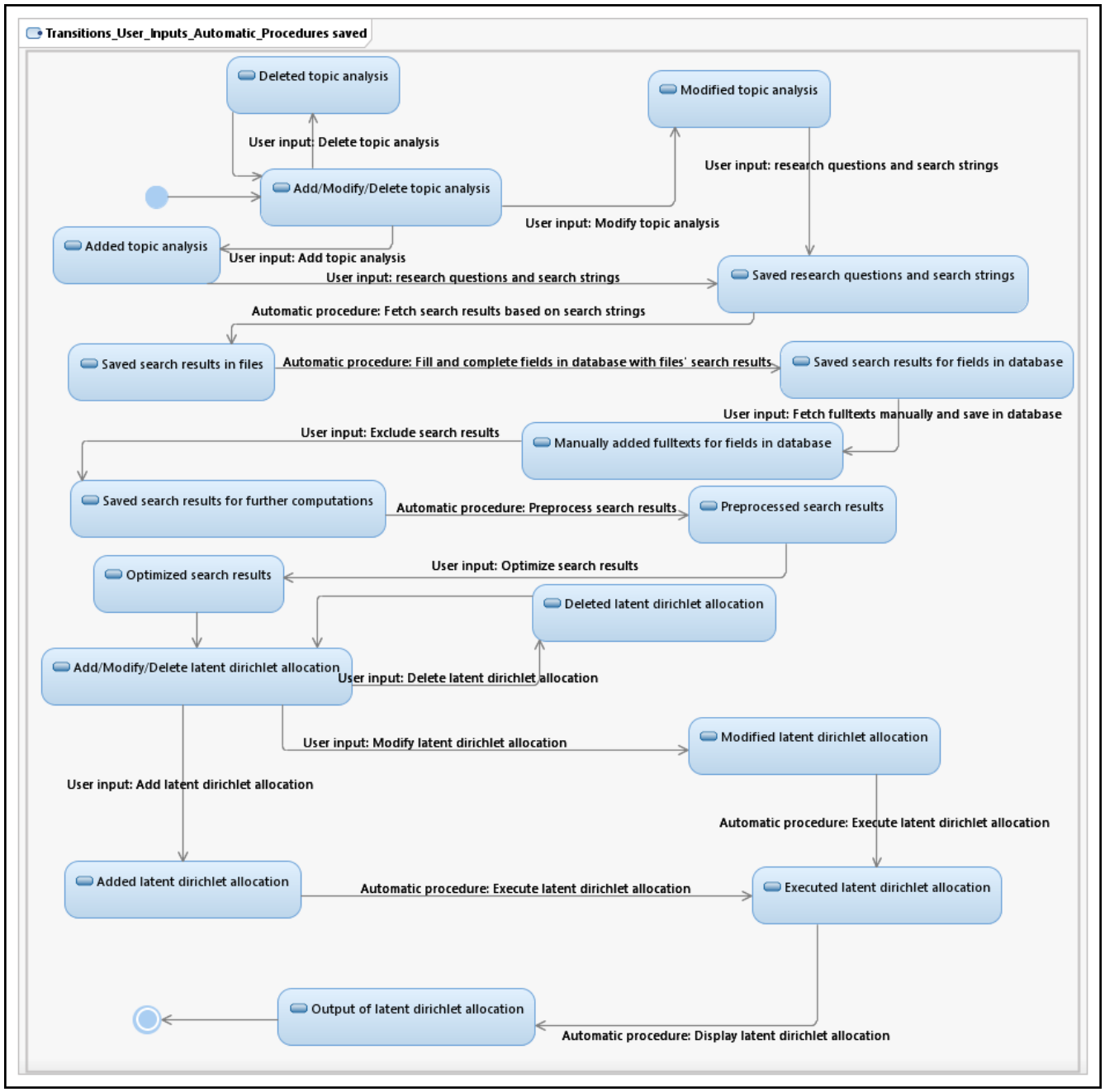


Figure 3.4 Dependency graph for functionalities that are part of the topic analysis tool created with IBM Rational Software Architect

3.3 Implementation

In this section the implementation of the functionalities in figure 3.4 in section 3.2.3 is presented as follows:

1. Adding, modifying and deleting a topic analysis environment consisting of a category.
   Any functionality that is a successor of the functionalities for adding, modifying and deleting a topic analyis environment is prevented from executing if no research questions and/or search strings exist for this topic analysis environment. This shall ensure a research question and search string based topic analysis based on the systematic mapping study according to [5, slide 12-19].

2. Collecting the search results in the topic analysis tool is the issue of the fourth and fifth functionality to be presented.

3. The sixth functionality is about excluding search results. Submitting the exclusion at least once for a topic analysis environment is mandatory in the sixth functionality otherwise any functionality after the sixth functionality cannot be used.

4. Between functionality seven and ten preprocessing is shown to prepare the abstracttexts and fulltexts for latent dirichlet allocation.

5. Preprocessing is a precondition for optimizing and as a last restriction is necessary for the execution of latent dirichlet allocation. This shall ensure that latent dirichlet allocation solely captures words as input that represent a selected natural language.

6. Functionality 11, 12 and 13 is about adding, modifying and deleting an environment for latent dirichlet allocation.

7. Functionality 14 copies a selected text corpus to an input folder for the execution of latent dirichlet allocation.

8. Functionality 15 is about displaying the result of an executed latent dirichlet allocation.

We choose flowcharts in order to show the features within the above enumerated functionalities. It is recommended to compare the content of these flowcharts with the database information provided in section 3.2.2 and with the source code provided by the DVD beside this diploma thesis.

3.3.1 Functionality for adding a topic analysis environment

This functionality is for adding a topic analysis environment for a category. First of all a category's name must be entered as a unique topic analysis environment's name. If the name is not unique then the topic analysis environment's name is not created. The procedural way for creating the category's name is illustrated in figure 3.5.

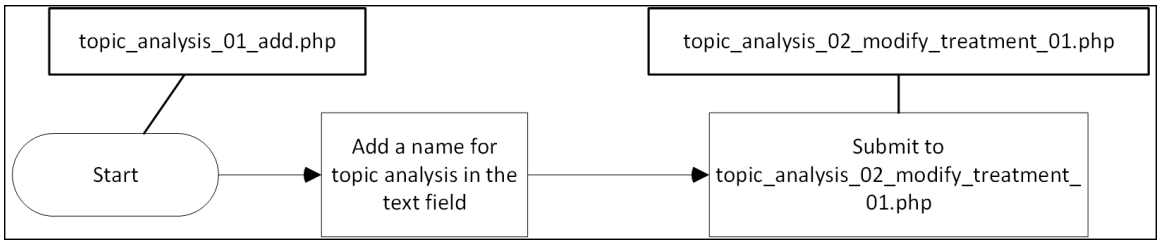


Figure 3.5 Flowchart for adding a category's name as a new topic analysis environment's name

If adding the category's name as a new topic analysis environment's name is successful then the research questions and the search strings must be entered. The flowchart for this procedural step is below in figure 3.6.

For saving the research questions and search strings the procedure illustrated in the flowchart in figure 3.7 is necessary.

3.3.2 Functionality for modifying a topic analysis environment

This functionality is for modifying a topic analysis environment for a category. First of all a category's name must be selected as a unique topic analysis environment's name. The procedure for selecting a category's name is illustrated in figure 3.8.

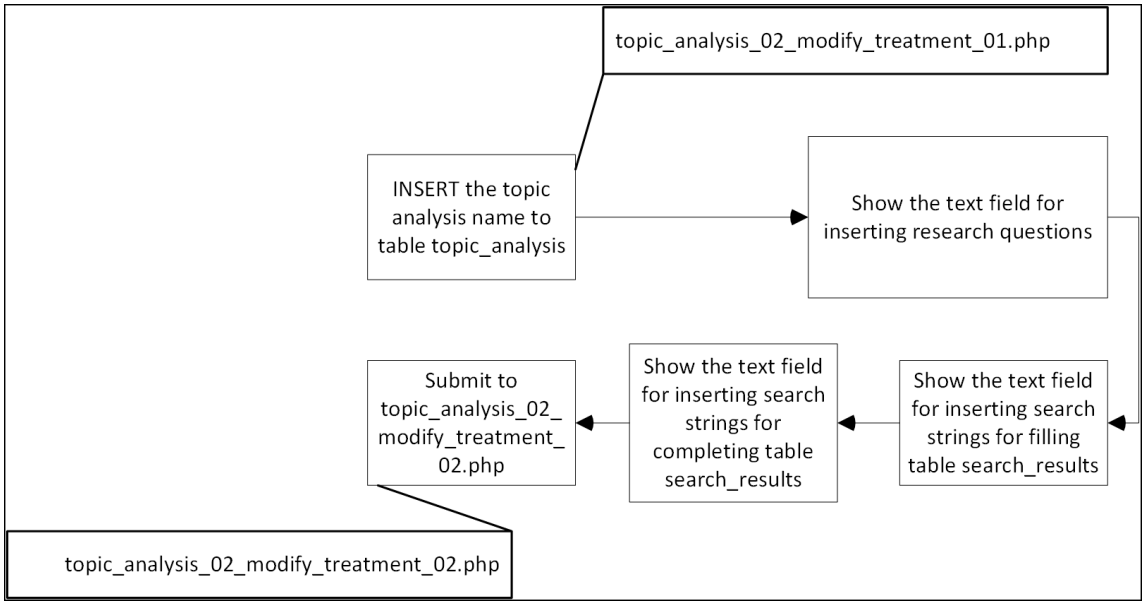


Figure 3.6 Flowchart for loading the form for saving the research questions and the search strings

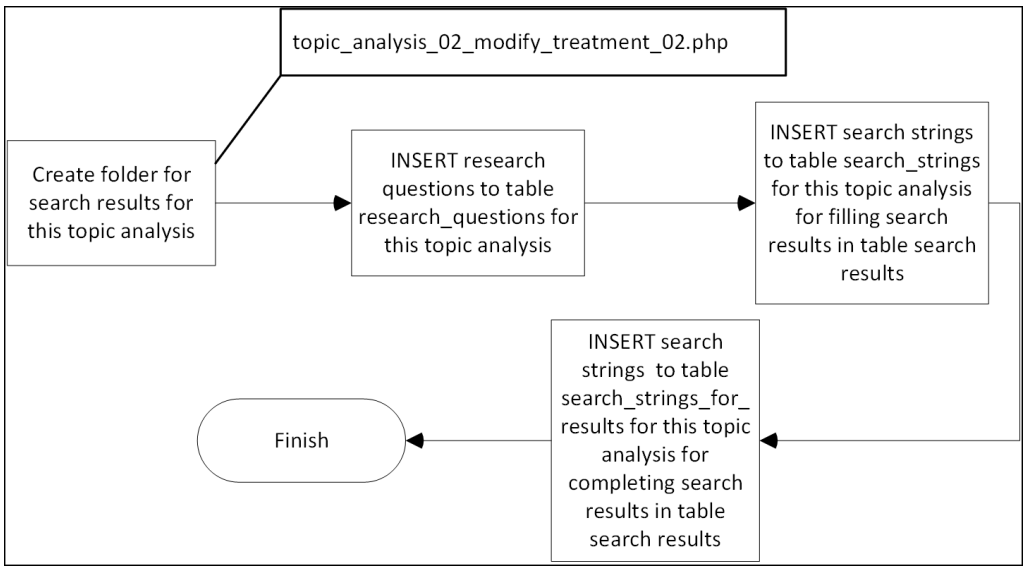


Figure 3.7 Flowchart for saving the research questions and the search strings for the category

If a topic analysis environment's name is selected then the topic analysis environment's name, the research questions and the search strings can be modified as illustrated in figure 3.9 and saved as illustrated in figure 3.10.

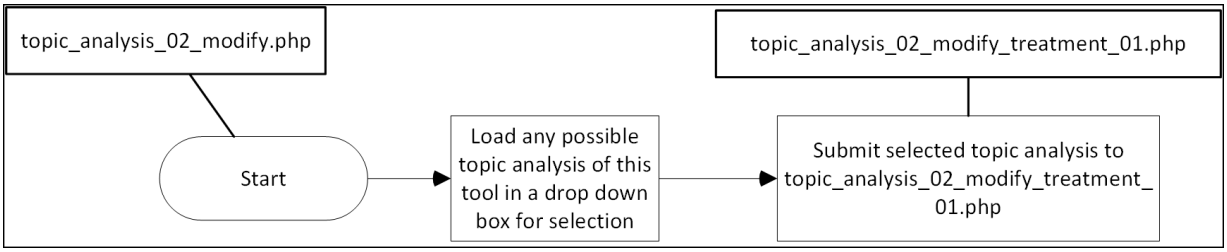


Figure 3.8 Flowchart for selecting a topic analysis environment's name

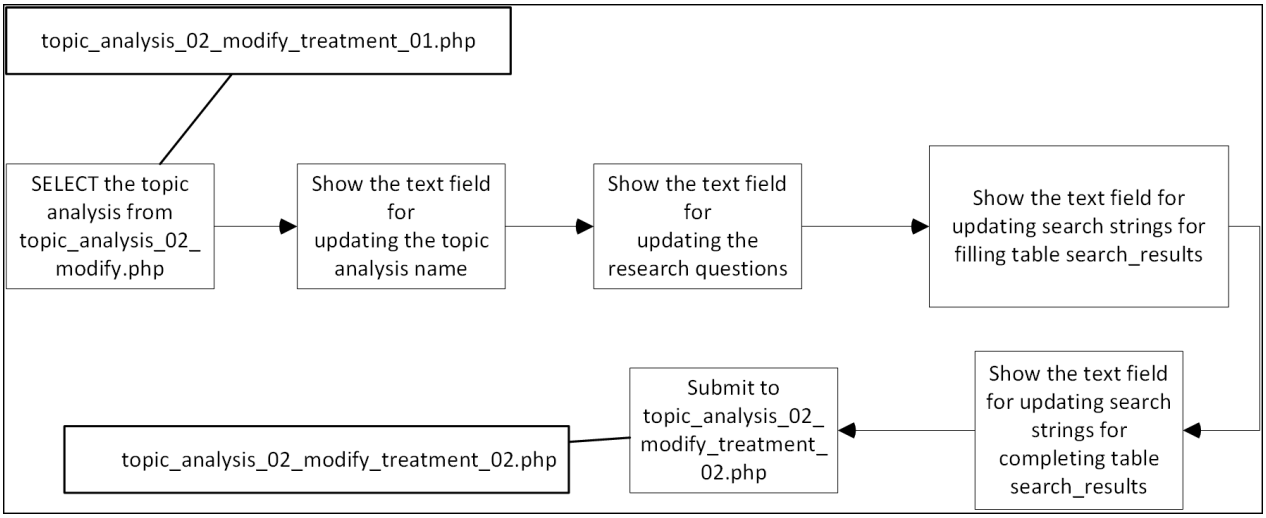


Figure 3.9 Flowchart for modifying the topic analysis environment's name, research questions and the search strings

3.3.3 Functionality for deleting a topic analysis environment

If the topic analysis environment contains a category that is not longer needed then this topic analysis environment can be selected in order to delete any folder, file and database entry for this topic analysis environment. For starting the procedure for deleting a topic analysis environment a topic analysis must be selected. This selection is possible with the help of the procedure illustrated in figure 3.11.

If the selected topic analysis environment is submitted then a the user has to confirm that the topic analysis environment should be deleted irrevocably. This procedure is shown in figure 3.12.

In case the deletion of a topic analysis environment is confirmed then the deletion is executed in the sequence which is illustrated in figure 3.13.

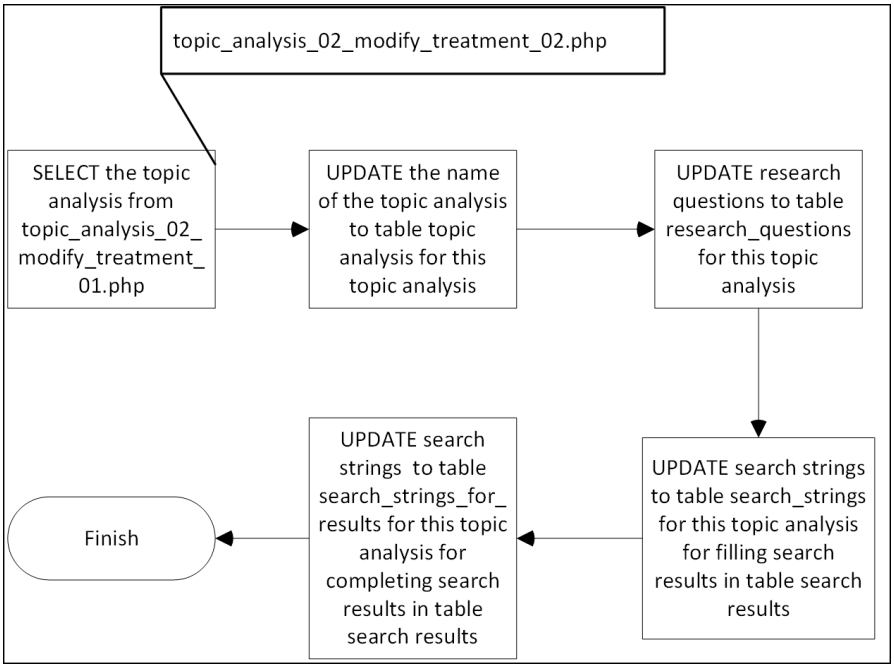


Figure 3.10 Flowchart for saving the topic analysis environment's name, the research questions and the search strings

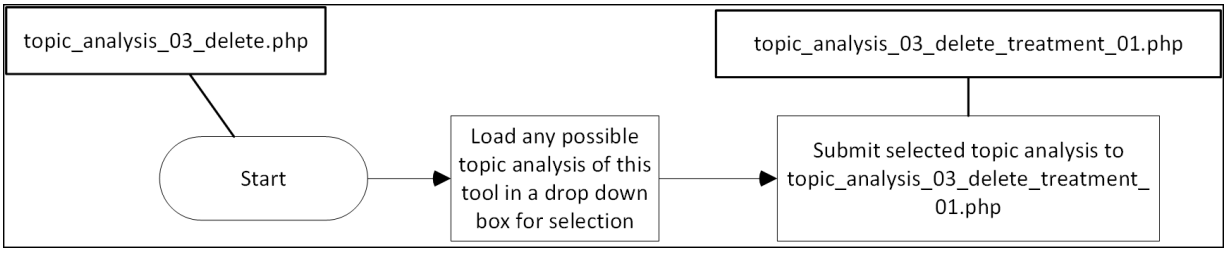


Figure 3.11 Flowchart for selecting a topic analysis environment's name

3.3.4 Collect search results for a topic analysis environment

This functionality is responsible for the information exchange between the web search engines that are addressed because of the usage of search strings added or modified in the functionality for adding or modifying a topic analysis environment. A first survey that shows the way information takes from the topic analysis tool to the requested web search engine and from the web search engine back to the topic analysis tool is illustrated in figure D.1. What web search engine to address depends on the URL of the search strings saved before. Web search

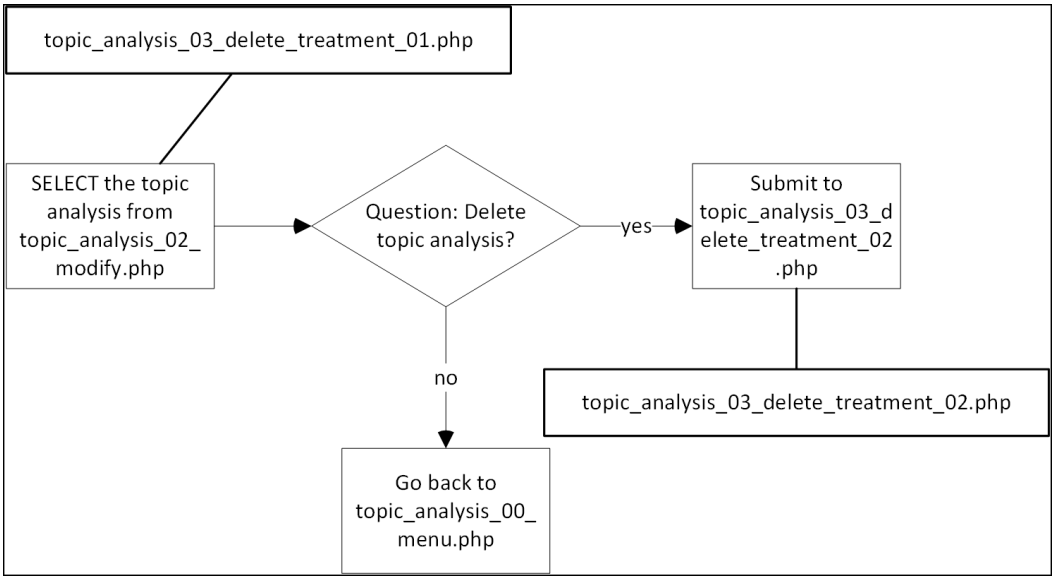


Figure 3.12 Flowchart for confirming the deletion of a topic analysis environment

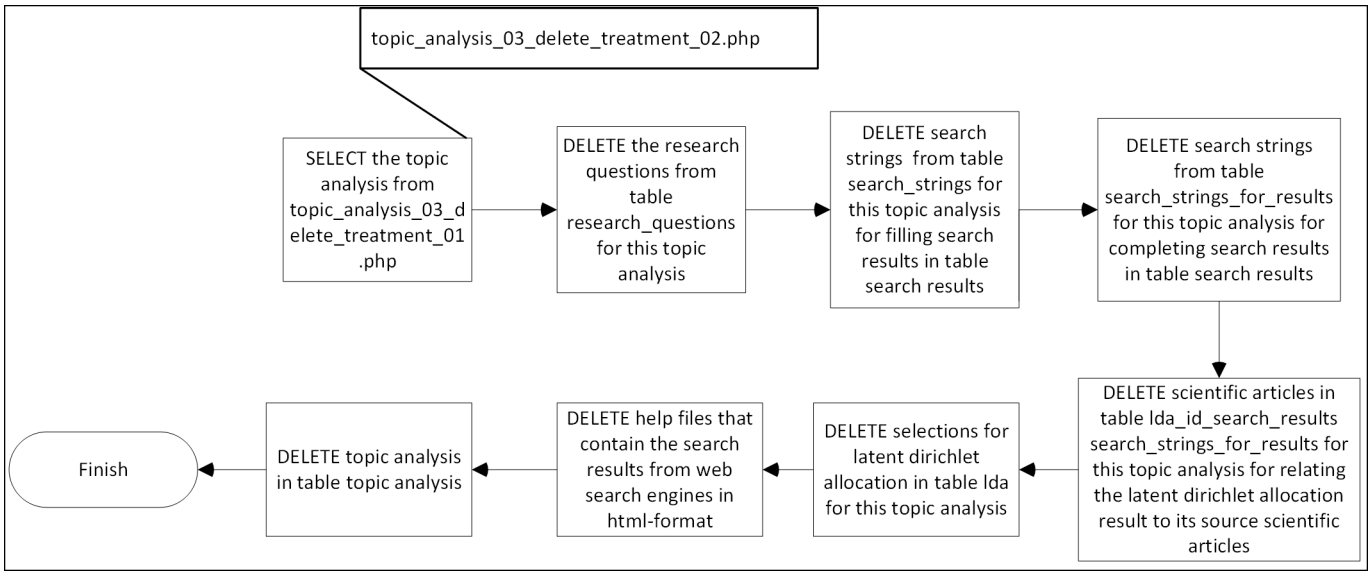


Figure 3.13 Flowchart for deleting a topic analysis environment

engines are considered that can return search results for as much scientific articles' attributes as possible which at the moment are "authors", "title", "year" and "conference", "abstracttext" and "fulltext". The procedure that has the task to fill as much attributes as possible with search results is called "filling-procedure". For attributes that remain without search results after the "filling-procedure"

the "completing-procedure" is regarded. The completing-procedure is a relation between handler identifier and the scientific articles attributes in table "search_results" as illustrated in figure 3.14

Figure 3.14 Relations between handler identifier and scientific articles' attributes in table "search_results"

In order to have a concept of the main feature fill and complete for this functionality the following features are introduced:

1. The filling-procedure for search results for the attributes "authors", "title", "year" and "conference" from the web search engine `http://dblp.org/search/publ/api`.

2. The completing-procedures for links to abstracttexts from `https://www.researchgate.net/search?q=`.

3. The completing-procedure for abstracttexts and links to pdffulltexts[4] from https://www.researchgate.net/.

4. The completing-procedure for path to pdffulltext.

5. The completing-procedure for pdffulltext as text.

6. The completing-procedure for pdffulltext as text extracted.

In order to start with point 1 the topic analysis tool has to pose a request for each search string in figure 3.15 with the help of "04_fill_and_complete/tools/wget.exe"[5] for filling table "search_results" with initial search results as follows:

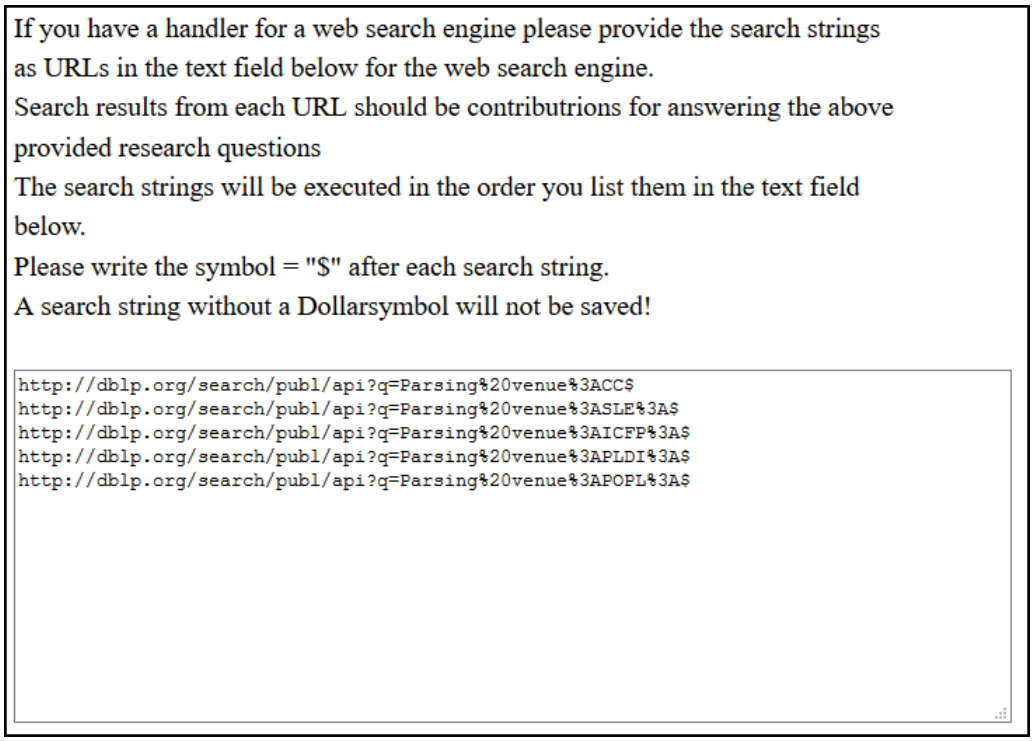


Figure 3.15 Search strings for the filling-procedure

[4] "pdffulltext" is the prefix or suffix for columnnames in table "search_results" which is equivalent to "fulltext".

[5] "wget.exe" is from http://www.gnu.org/software/wget.

Let "wget.exe" save the search results from http://dblp.org/search/publ/api for each search string in figure 3.15. Use the handler for this filling-procedure for http://dblp.org/search/publ/api[6] and save the values from the search results from http://dblp.org/search/publ/api to the scientific articles' attributes "authors", "title", "year" and "conference"[7].
In terms of referring to point 2 from above the completing-procedure for fetching the links to abstracttexts from https://www.researchgate.net/search?q= starts with fetching a title from any scientific article in table "search_results" that belongs to a particular topic analysis environment. Then https://www.researchgate.net/search?q= is concatenated with each "title" of the scientific articles of this topic analysis environment in order to download the search results for each https://www.researchgate.net/search?q=[$title] with the help of "wget.exe". In the next step the links to the abstracttexts at https://www.researchgate.net/ are extracted to the column "first_link_to_abstracttext" in table "search_results" of each scientific article whose found "title" is equal to the "title" of researchgate's search result[8,9].
In terms of referring to point 3 from above the next completing-procedure is the completion-handler for fetching the abstracttexts from https://www.researchgate.net/. Moreover, we will extract the links to pdffulltexts from the web search engine's search results if pdffulltexts are available in these search results.
This completion-handler starts with fetching the scientific articles' values from the scientific articles' attribute "first_link_to_abstracttext". We fetch scientific articles' values from the attribute "first_link_to_abstracttext" at first because it is characteristic for https://www.researchgate.net/ that abstracttexts are not retrievable with the help of the scientific article's title in the URL equal to https://www.researchgate.net/search?q=[$title]. We need the scientific articles' values from from the attribute "first_link_to_abstracttext" for fetching the full abstracttexts from https://www.researchgate.net/. Thus any value from "first_link_to_abstracttext" from table "search_results" from this topic analysis environment is used for a http-request with the help of "wget.exe".

[6]The handler for point 1 is visualized in figure D.8 and in figure D.9.
[7]The flowchart for point 1 is visualized in figure D.2.
[8]The handler for point 2 is visualized in figure D.10
[9]The flowchart for point 2 is visualized in figure D.3.

The search results that "wget.exe" creates are scanned for the abstracttext and saved as a value in the field for the scientific article's attribute "abstracttext" where the title of the search result is the same as the title of the found scientific article in table "search_results". If a link to a fulltext is found in the search result of "wget.exe" then it is also saved to the found scientific article in attribute "link_to_pdffulltext" of table "search_results"[10].
We continue with point 4 from above. This procedure's task is to complete the scientific articles' attribute "path_to_pdffulltext" of table "search_results" of already downloaded fulltexts with the help of the URLs in the filled and above mentioned attribute "link_to_pdffulltext". Thus we download the fulltexts with the help of "wget.exe" and save the local fullpath to each fulltext in the attribute "path_to_pdffulltext" of table "search_results"[11].
The next completing procedure according to point 5 from above is a conversion from any downloaded fulltext with the local fullpath in "path_to_pdffulltext" to text in "pdffulltext_as_text" with the help of "04_fill_and_complete/tools/pdftotext.exe"[12,13].
The last completion handler according to point 6 from above is for manual adaption of any fulltext in "pdffulltext_as_text" in table "search_results" to the needs of latent dirichlet allocation. Thus superfluous information like author-names or bibliography can be manually deleted by the user in this completion step. The results for the manual change after submit are copied to "pdffulltext_as_text_extracted" in table "search_results"[14].

3.3.5 Manual upload of fulltexts for a topic analysis environment

After having finished the functionality "04_fill_and_complete" for filling and completing the topic analysis tool with scientific articles' values for scientific articles' attributes in table "search_results" it is possible that the completing-procedure could not save enough fulltexts to the scientific articles' attribute "pdffulltext_as_text_extracted" to satisfy the user's needs. For that reason this func-

[10] The flowchart for point 3 is visualized in figure D.4.
[11] The flowchart for point 4 is visualized in figure D.5.
[12] "pdftotext.exe" is from http://www.foolabs.com/xpdf/ at 2017-09-30.
[13] The flowchart for point 5 is visualized in figure D.6.
[14] The flowchart for point 6 is visualized in figure D.7.

tionality gives the user the possibility to manually upload fulltexts with the goal that texts of these fulltexts are saved as a value to the scientific articles' attribute "pdffulltext_as_text_extracted" of the focused scientific article.
This upload is possible with the help of a snapshot which is created after the topic analysis environment is selected.
The selection of the topic analysis environment is illustrated in figure 3.16.

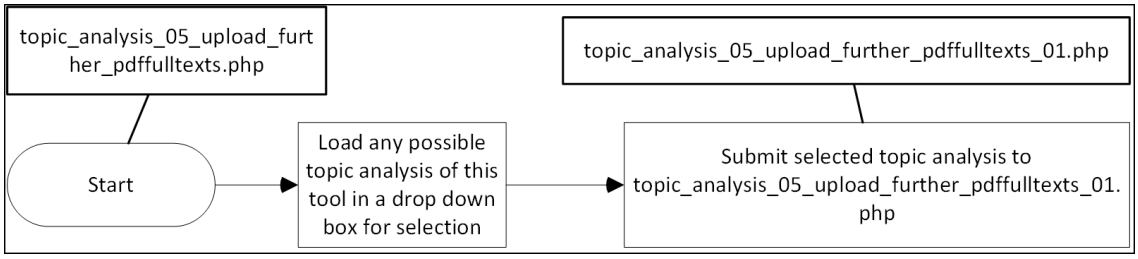


Figure 3.16 Flowchart for selecting a topic analysis environment's name

After the topic analysis environment is selected the search results that belong to this topic analysis environment can be loaded into the snapshot. The snapshot shows an input field for the scientific article's attribute "path_to_pdffulltext" whose scientific article as a row in table "search_results" is without a fulltext. If the user has a fulltext to upload then the internet's http-address (possibly of a public home directory of the user) to the fulltext in the must be entered into the input field of a focused scientific article[15]. If the fulltext is successfully uploaded then the text of the fulltext is extracted to the scientific articles' attribute "pdffulltext_as_text_extracted" for the focused scientific article. Also the "path_to_pdffulltext" is updated.
A flowchart for showing the snapshot is presented in figure 3.17.
A flowchart for updating "path_to_pdffulltext" and "pdffulltext_as_text_-extracted" is shown in figure 3.18.

3.3.6 Exclude search results for a topic analysis environment

The exclusion of search results, i.e. scientific articles where each scientific article is a row in table "search_results" is possible if the value of a checkbox of a scien-

[15]This internet's http address must be reachable by the topic analysis tool otherwise the upload of the selected fulltext will fail.

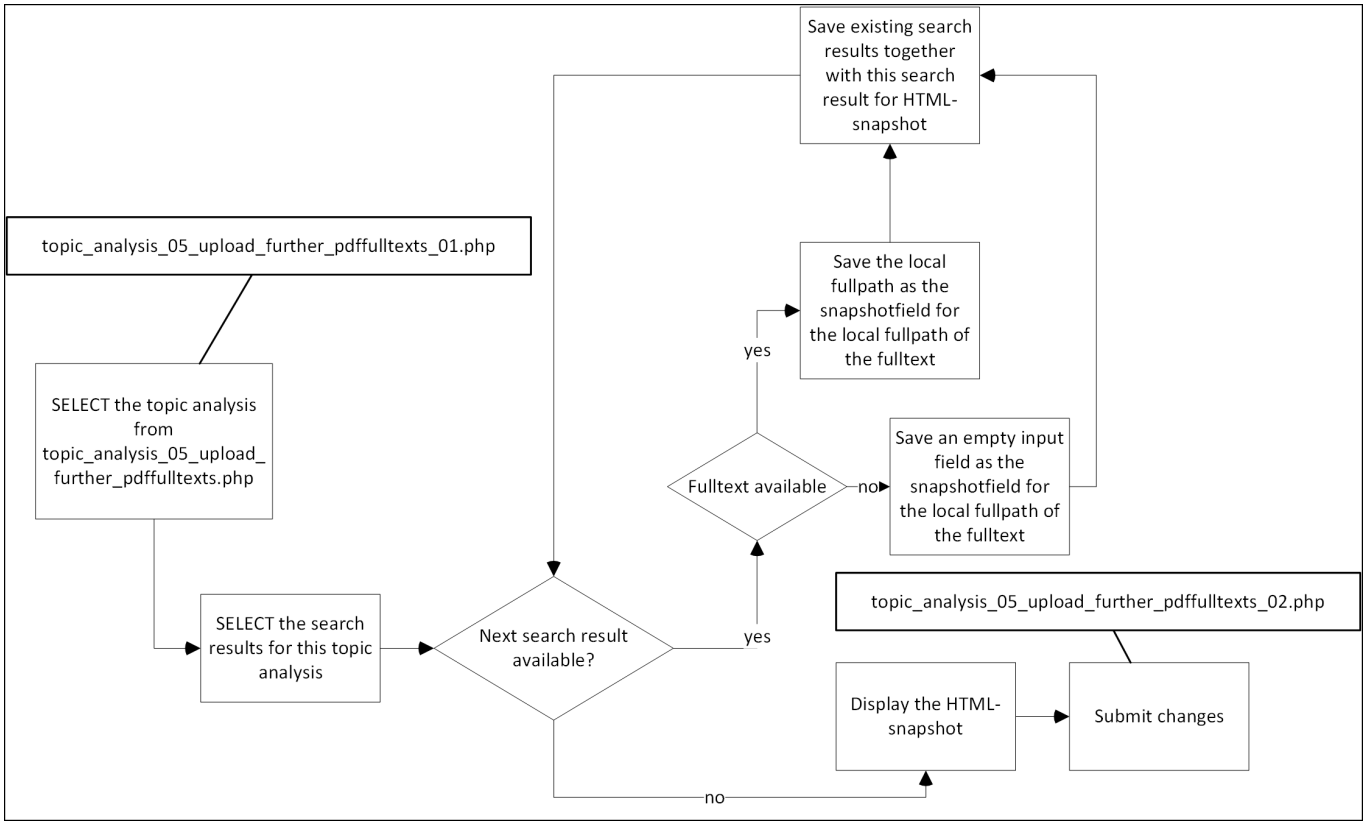


Figure 3.17 Flowchart for displaying the snapshot

tific article in its attribute "exclude" is checked.
In order to load the table "search_results" with scientific articles for the selected topic analysis environment the procedure illustrated in figure 3.19 must be submitted. The procedure illustrated in figure 3.20 then loads the snapshot with the possibility to exclude. If the procedure in figure 3.20 is submitted then the value "1" is updated to each field of the attribute "exclude" that is checked in the procedure before. Otherwise this exclude-value is "0". Each scientific article with an updated value for its attribute "exclude" is marked with the update value "1" in its attribute "exclusion_already_done". This is because according to the systematic mapping study in [5, slide 20-22] the exclusion procedure is a prerequisite for further topic analysis with this tool. We ensure with the marker in attribute "exclusion_already_done" that exclusion happened before. If this is the case then the not excluded scientific articles are provided for the further functionalities of

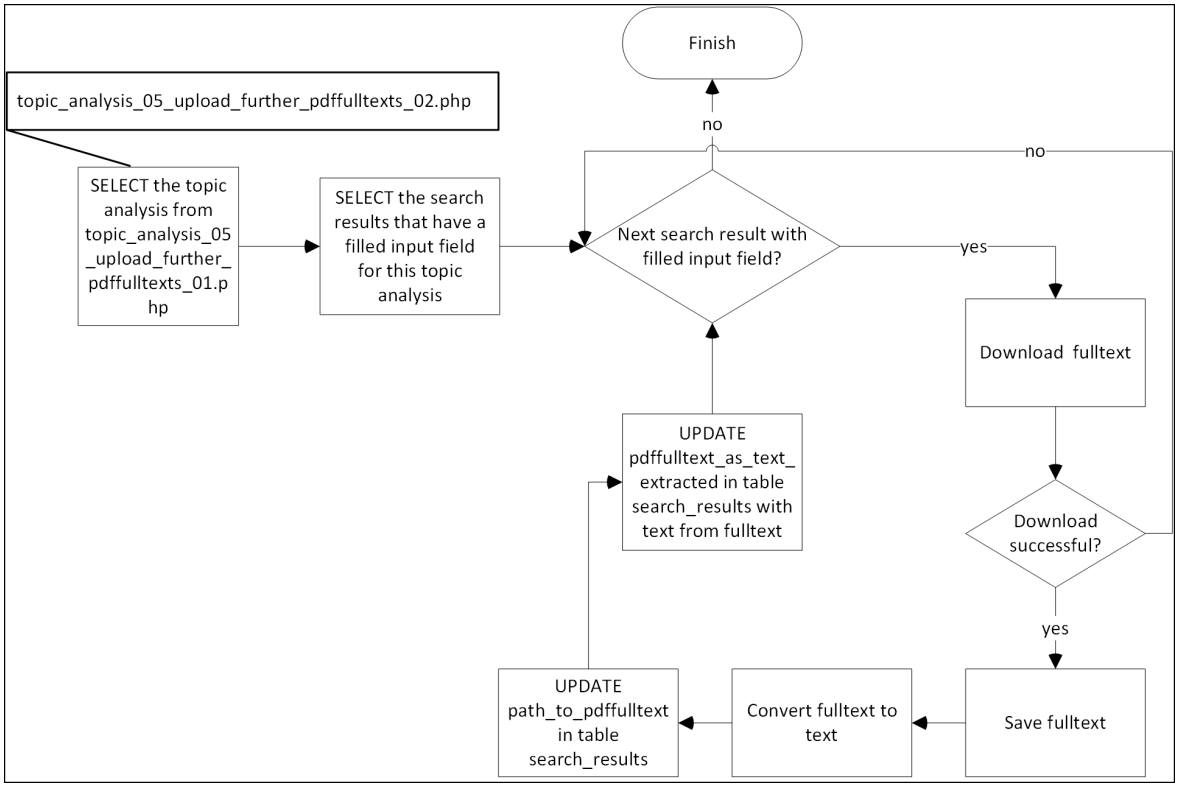


Figure 3.18 Flowchart for updating table "search_results"

this topic analysis tool.
Any update to table "search_results" is illustrated in figure 3.21.

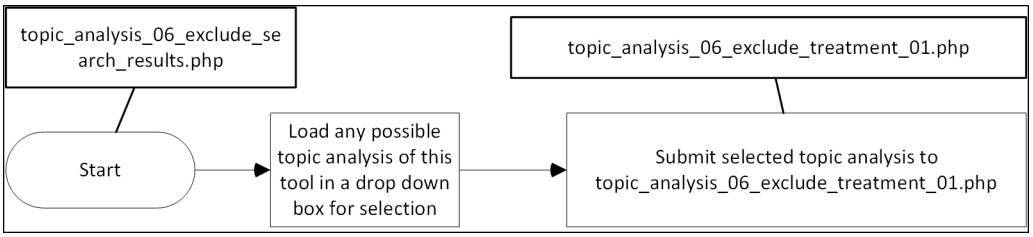


Figure 3.19 Flowchart for selecting a topic analysis environment's name

3.3.7 Preprocessing abstracts and fulltexts for a topic analysis environment

In this section the functionality that preprocesses the input of the execution of latent dirichlet allocation is described. The input for preprocessing is from

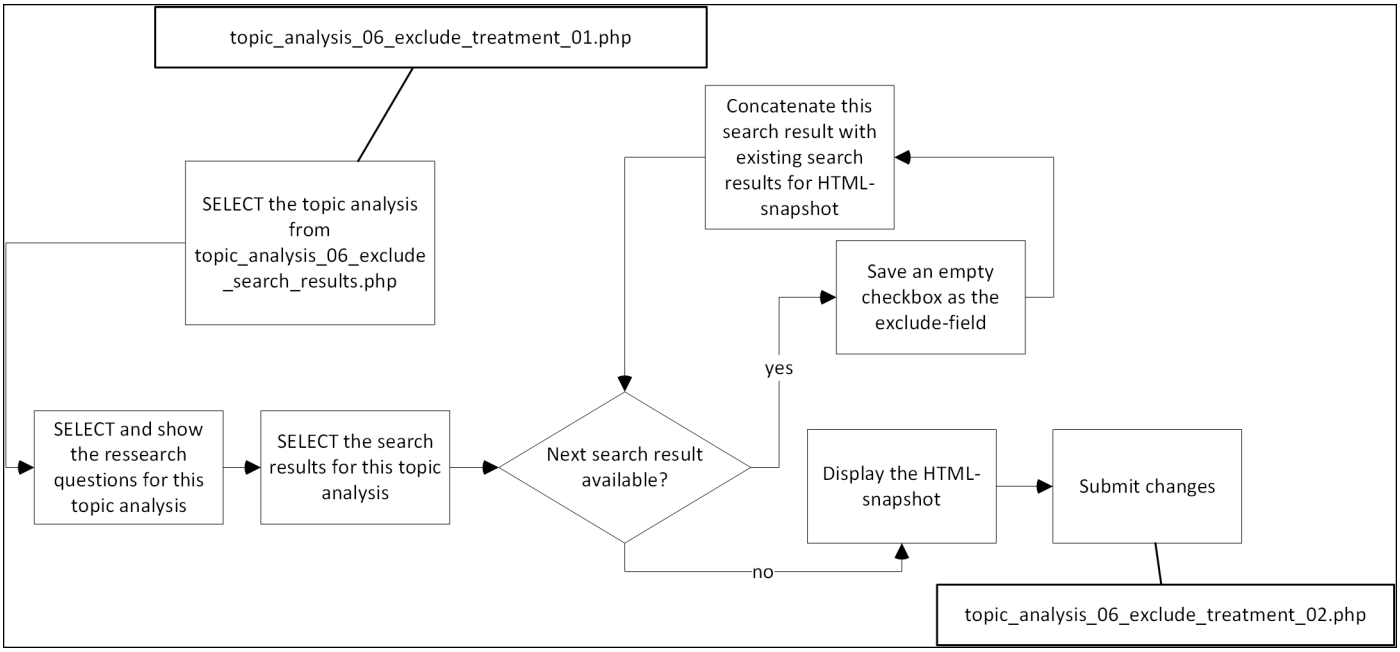


Figure 3.20 Flowchart for displaying the snapshot

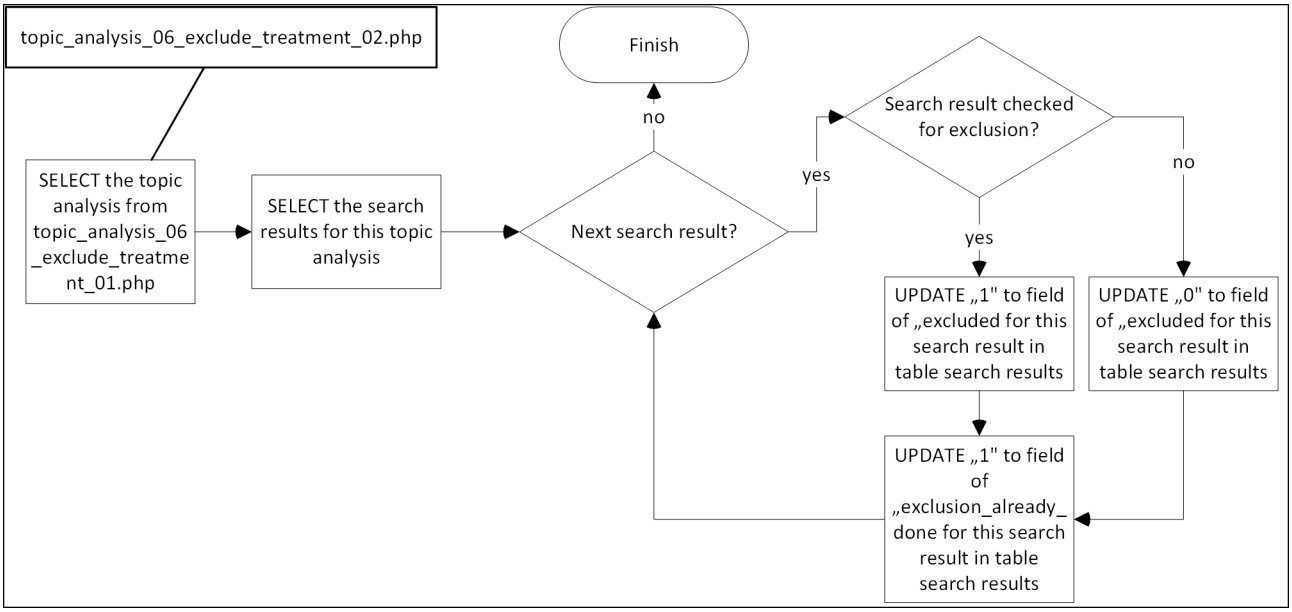


Figure 3.21 Flowchart for updating table "search_results"

abstracttexts from scientific articles' attributes "abstracttext" and from full-texts from the scientific articles' attribute "pdffulltext_as_text_extracted" in table "search_results".

The main part of the preprocessing procedure is that we select any input in order to analyze for each text character of this input whether the text character is

- a-z,
- A-Z,
- fullpoint,
- space,
- linefeed or
- carriage return.

If the inspected text character is equal to one of the above listed text characters then this text character is concatenated with existing text characters in a temp-variable. The content of this temp-variable is updated to the field for abstracttext in "abstracttext_for_lda" or to the field for fulltext in "fulltext_for_lda" depending on what input is processed. Values in these two scientific articles' attributes are the input values for latent dirichlet allocation whose preparation is described in section 3.3.10.
The procedures for processing are fulltexts are identical with the procedures for processing abstracttexts. For that reason any procedure step for preprocessing fulltexts is illustrated between figure 3.22 and figure 3.23 as well as for abstracttexts[16].

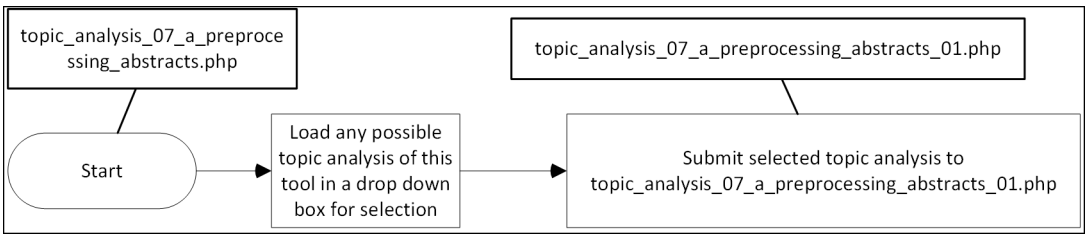


Figure 3.22 Flowchart for selecting a topic analysis environment's name

[16]For fulltext the word "abstracttext" must be replaced by the word "pdffulltext" in case of table field names. Otherwise "abstracttext" must be replaced by "fulltext" in the illustrations between figure 3.22 and figure 3.23.

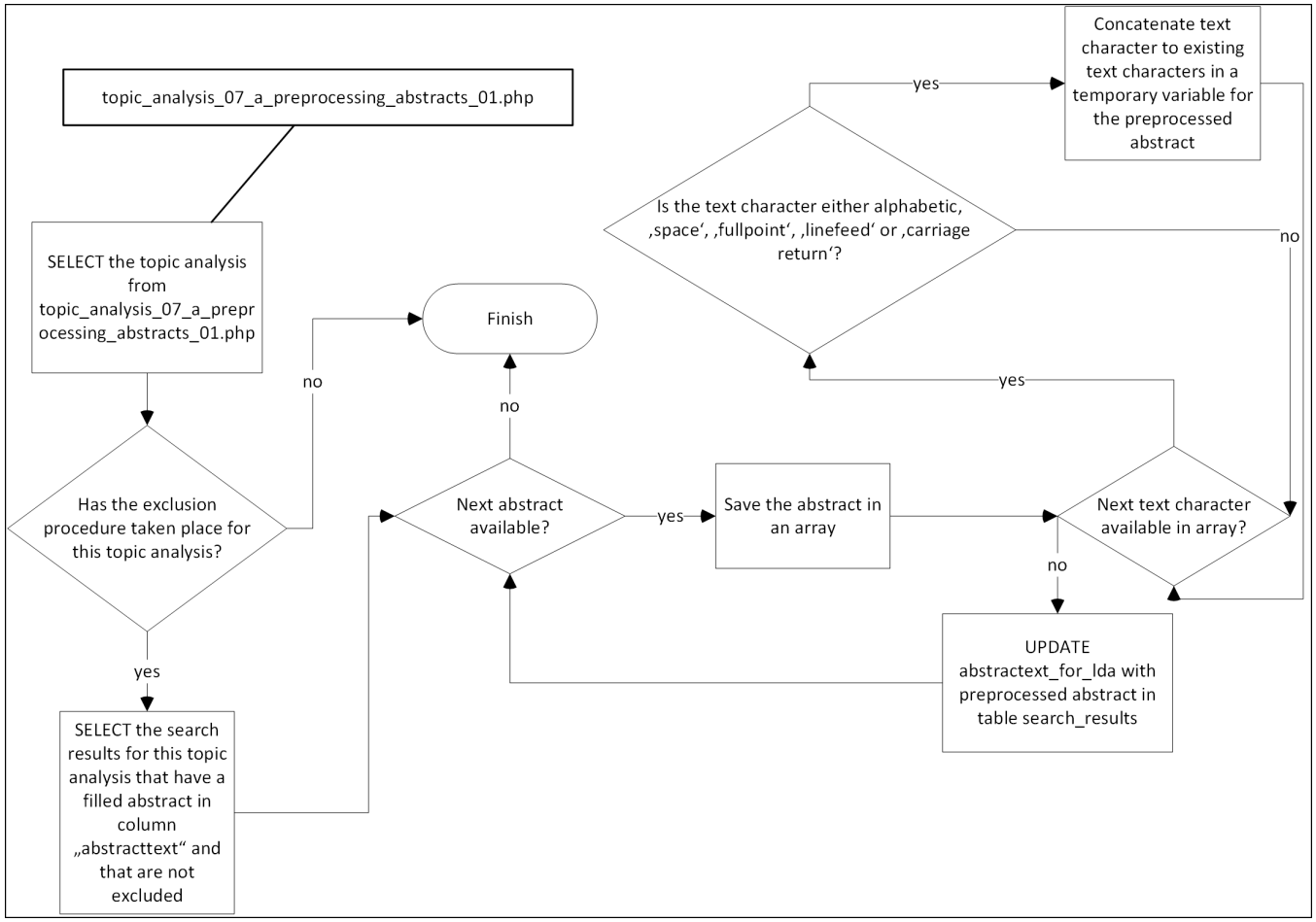


Figure 3.23 Flowchart for the preprocessing procedure for abstracttexts

3.3.8 Optimizing abstracts and fulltexts for a topic analysis environment

In order for manual exclusion of words that are not part of the selected natural language and, moreover, that do not contribute the study and summary of a category for which a topic analysis environment is created for[17], optimizing should be executed. In order to have a survey of the procedure of optimizing abstracttexts and fulltexts the flowcharts for abstracttexts are illustrated between figure 3.24 and figure 3.26. There are just flowcharts for abstracttexts shown because

[17]Studying and summarizing a topic analysis environment's category is demanded in section 2.1 in terms of the systematic mapping study and in section 2.3 in terms of topic analysis.

the procedures of abstracttexts and fulltexts are identical in terms of optimizing. For fulltext the word "abstracttext" must be replaced by the word "pdffulltext" in case of table field names. Otherwise "abstracttext" must be replaced by "fulltext".

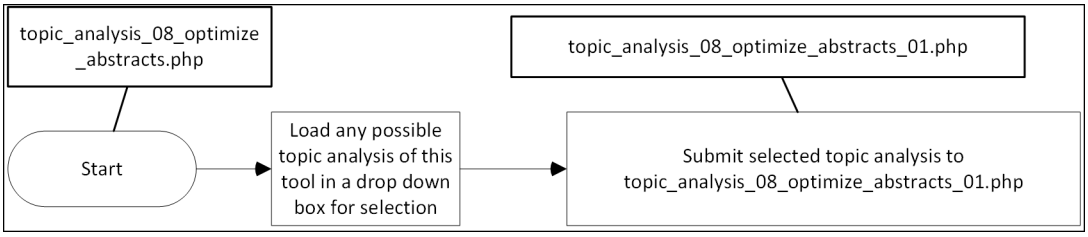


Figure 3.24 Flowchart for selecting a topic analysis environment's name

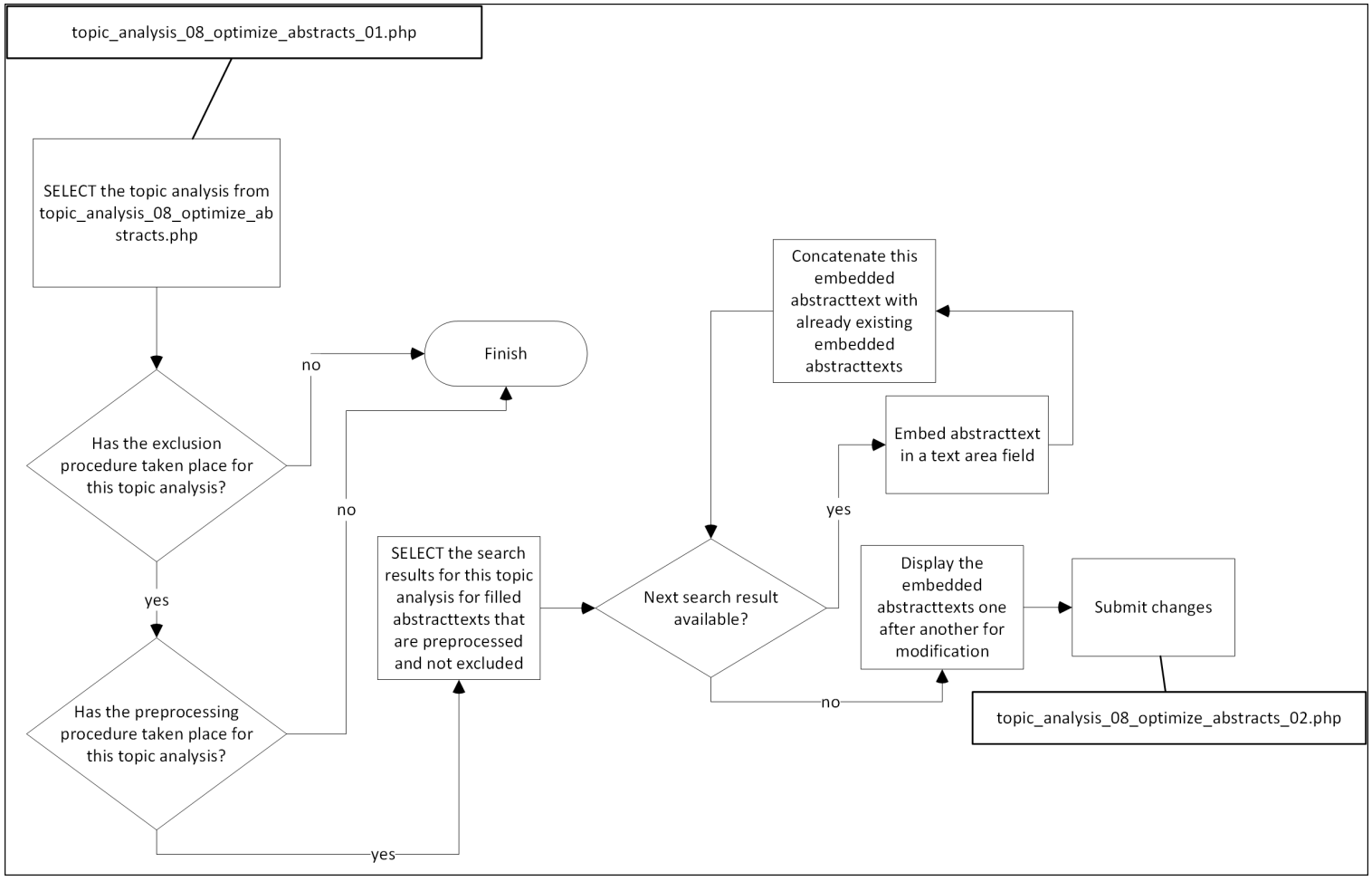


Figure 3.25 Flowchart for showing the abstracttexts to optimize

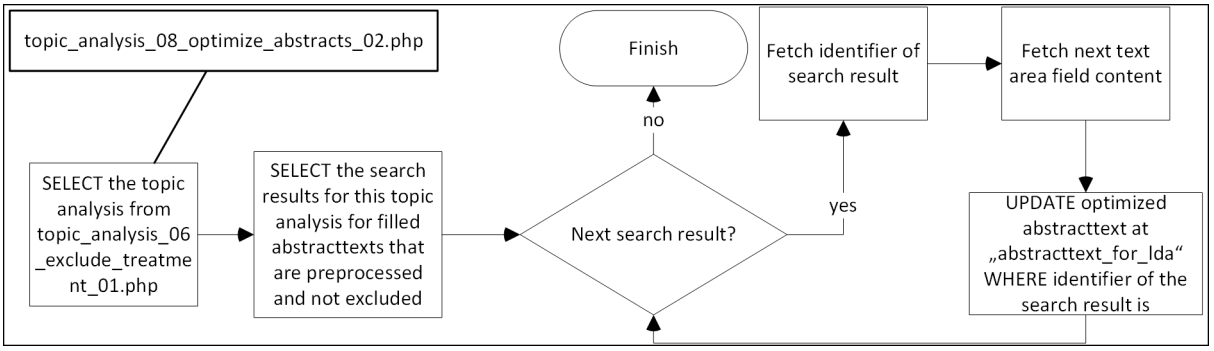


Figure 3.26 Flowchart for saving the optimized abstracttexts to table "search_results"

3.3.9 Add and modify an environment for latent dirichlet allocation

An environment for latent dirichlet allocation that is created because of purposes that are described in section 4.11 consists of the following procedures:

1. Select a topic analysis environment.
2. Show the input form for adding a new latent dirichlet allocation environment.
3. Save the values from the form for adding a new latent dirichlet allocation environment.

We give a survey for each listed item above in the flowchart in figure 3.27 and in figure 3.28 where the first flowchart is about any procedure step which is a prerequisite for saving the input parameters according to the description in section 4.11.

The saving script fulfills the following tasks:

- It creates a folder structure for the input-, output- and execute-files for this latent dirichlet allocation environment.
- It creates an executable file for the execution of latent dirichlet allocation for this latent dirichlet allocation environment with the help of templates and the information how much topics to create, where the input folder and where the output folder is for this latent dirichlet allocation environment[18].

[18] An example for a created execute-file for a latent dirichlet allocation environment named "CC" is shown in the appendix section E.1.

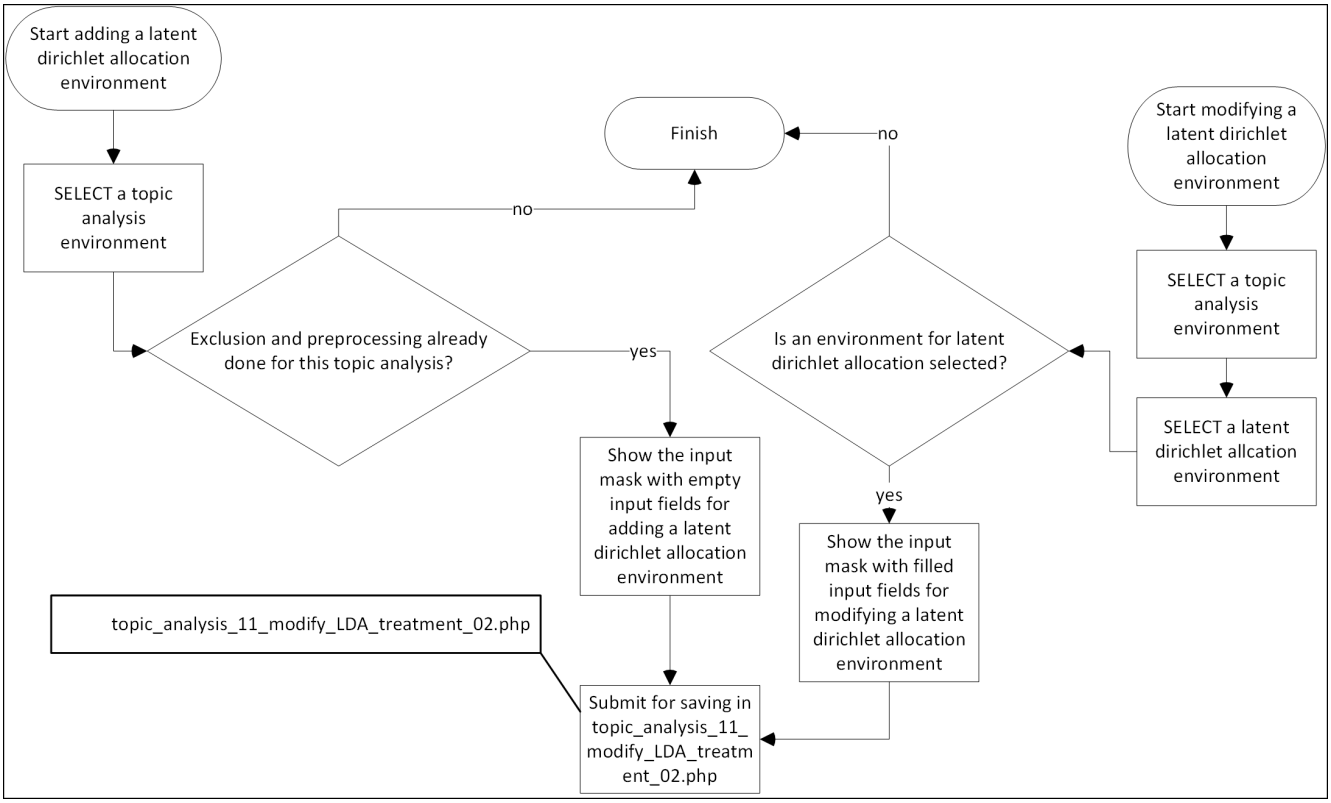


Figure 3.27 Flowchart: Any prerequisite procedure step for saving the input parameters of a latent dirichlet allocation environment

- It inserts the input parameters from the add- or modify-form and the string for the directory for input, output and execute-folders for this latent dirichlet allocation environment in table "lda".

However, constraints exist that could prevent the saving script from fulfilling these three tasks. Therefore, in order to prevent these constraints from coming into effect it is necessary that a topic analysis environment is selected before loading this script. This topic analysis environment must contain at least one search string which enables a selection of search results in table "search_results". If the constraint for a selection in table "search_results" is true that at least one search result is not excluded AND NOT (an abstracttext AS WELL AS an fulltext is not preprocessed) then the saving script does not break the execution but check whether the input parameter for the input dirichlet allocation execution of this tool are entered correctly.

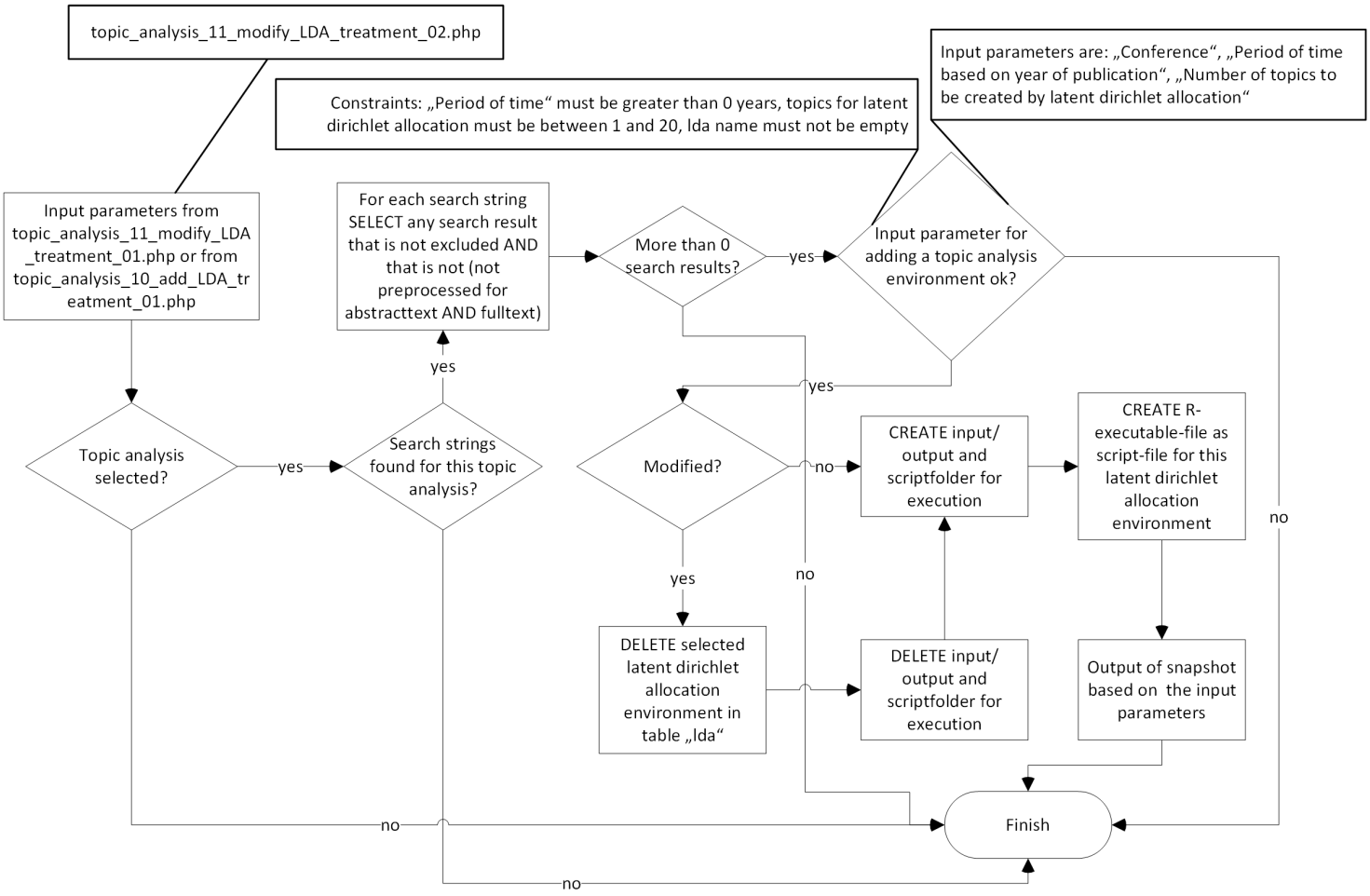


Figure 3.28 Flowchart: The saving script for adding and modifying a latent dirichlet allocation environment

3.3.10 Execute and delete an environment for latent dirichlet allocation

The prerequisite for executing latent dirichlet allocation for an environment for latent dirichlet allocation is the “preparation procedure” which is described in this section. The “preparation procedure” fetches the following entities as input:

1. The topic analysis environment that is selected first before entering the execution script.

2. The latent dirichlet allocation environment that is selected after the topic analysis environment.

3. At least one search string that is necessary to fill the table "search_strings"[19] and to relate table "topic_analysis" with table "search_results"[20].

4. The selections for "conference", "year from" and "year to" from the form for adding or modifying a latent dirichlet allocation environment.

Based on this provided input compiled by the information enumerated above the "preparation procedure" creates the following output that is the input for the execution of latent dirichlet allocation for this selected latent dirichlet allocation environment:

1. A selection of fulltexts or abstracttexts if particular fulltexts do not exist which is saved as input files in the input folder for the latent dirichlet allocation environment.

2. A selection of identifier for search results that related to the identifier of the selected latent dirichlet allocation environment are inserted in table "lda_id_search_results"[21].

How this "preparation procedure" transforms the values to input for the execution of the execution of latent dirichlet allocation can be described as follows: First of all the input data based on "conference", "year from" and "year to" is read in order to compute what input data is filled with which values. The result is the computation of the SQL-query that selects the scientific articles that are included in this selected values for "conference", "year from" and "year to" in table "search_results". The results are written as files with fulltexts or abstracttexts, if particular fulltexts do not exist, in the input folder of the selected latent dirichlet allocation environment. The results are also written in table "lda_id_search_results" as identifiers of the search results that contain these selected fulltexts or abstracttexts.

[19] Compare with the description for the functionality "04_fill_and_complete" in section 3.3.4.

[20] Compare with the E-R-model of the database of the topic analysis tool in figure 3.3 in section 3.2.2.

[21] These entries in table "lda_id_search_results" are needed for displaying the scientific articles that affect the output of the execution of latent dirichlet allocation for a selected latent dirichlet allocation environment. This issue is treated in section 3.3.11.

If this task is finished then the user is noticed with the message that informs how to start the execution of latent dirichlet allocation in R on these input files and the execution file created in line of adding or modifying a latent dirichlet allocation environment described in section 3.3.9[22].
The possibilities the "preparation procedure" has in order to compute the correct selection for latent dirichlet allocation execution input is illustrated in figure 3.29.

	Conference	From year	Until year
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

Figure 3.29 Condition cases for "conference", "year from" and "year to"

The transformation from input to output as described above is illustrated in the flowcharts between figure 3.30 and figure 3.31.

If the deletion procedure is called then the

- input-, output- and execution-folder,
- any entry in table "lda" and
- any entry in table "lda_id_search_results"

is deleted for the selected latent dirichlet allocation environment.

[22]How the execution is started, run and successfully finished is described in section 4.12 for a latent dirichlet allocation environment called "CC".

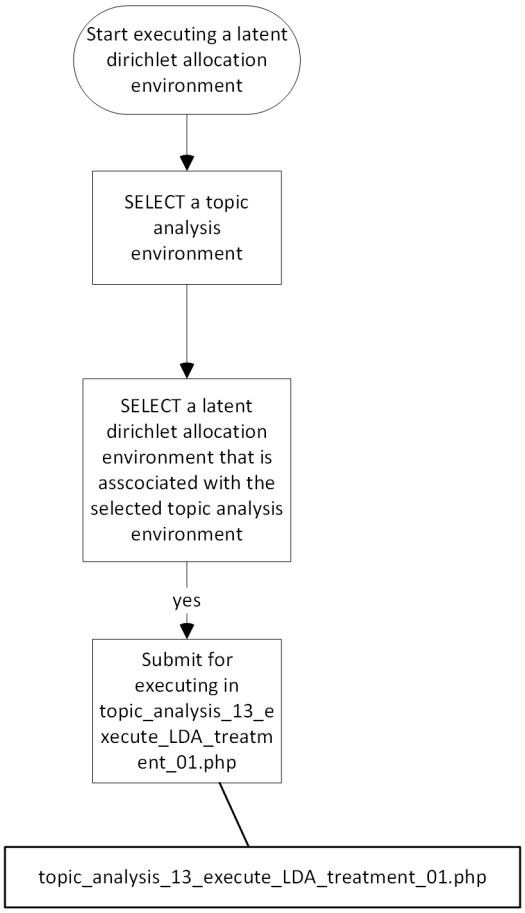


Figure 3.30 Flowchart: Select a latent dirichlet allocation environment for execution

3.3.11 Output of an executed latent dirichlet allocation environment

The output of an executed latent dirichlet allocation environment is retrievable with the help of the functionality "14_output_LDA". At first the topic analysis environment must be selected because the output is related to an environment for latent dirichlet allocation which is related to a topic analysis environment. The selection of a topic analysis environment is illustrated in a flowchart in figure figure 3.32.

A latent dirichlet allocation environment which is related to the topic analysis environment illustrated in terms of its selection in terms of figure 3.32 must be selected next which is illustrated in figure 3.33.

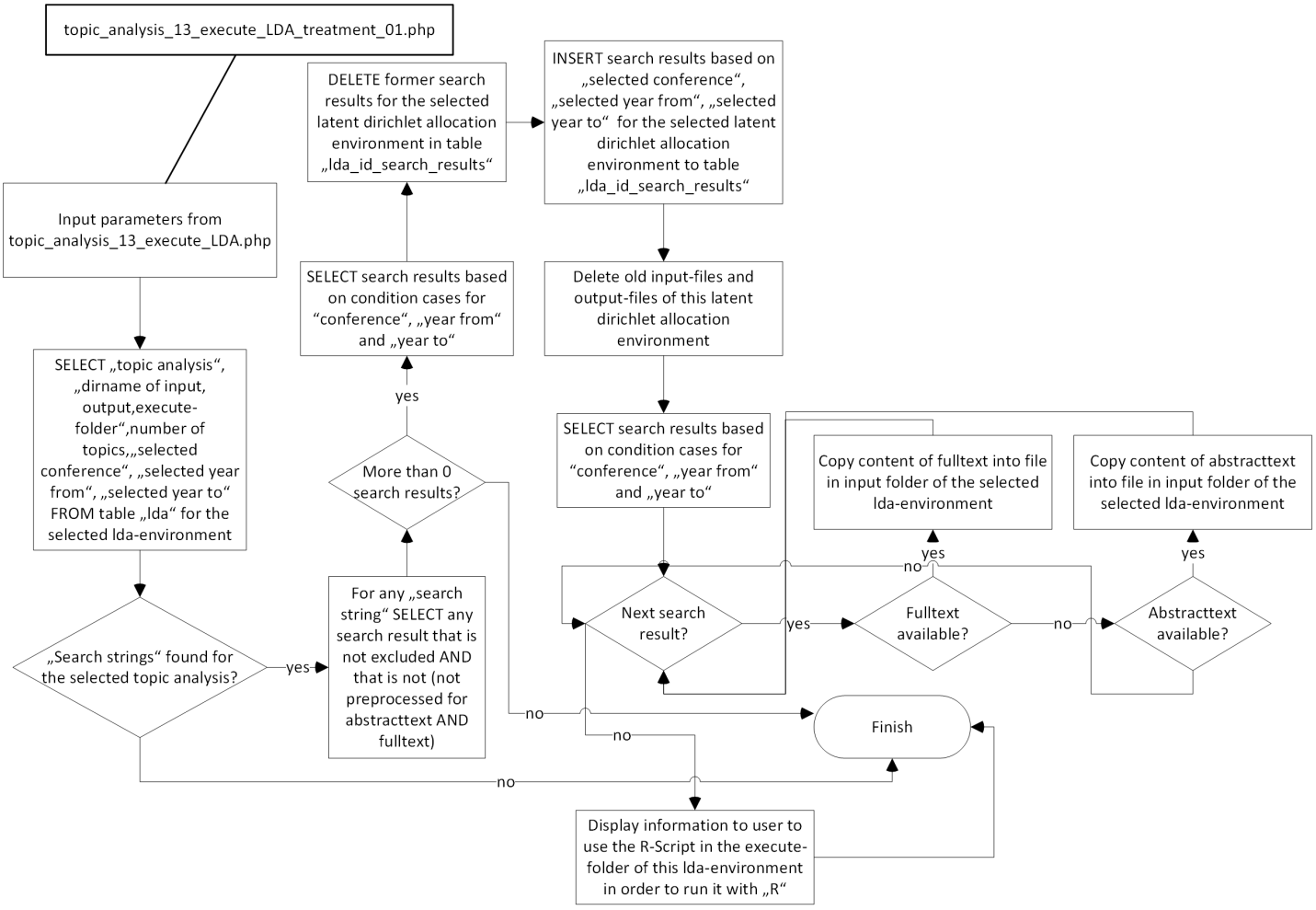


Figure 3.31 Flowchart: Execution of a latent dirichlet allocation environment

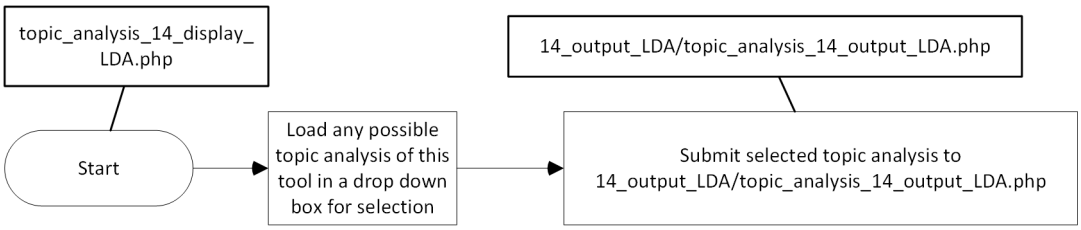


Figure 3.32 Flowchart: Selection of a topic analysis environment

After submitting the selection of the latent dirichlet allocation environment the output page for the selected latent dirichlet allocation environment is loaded because the output page is provided by the latent dirichlet allocation environment's

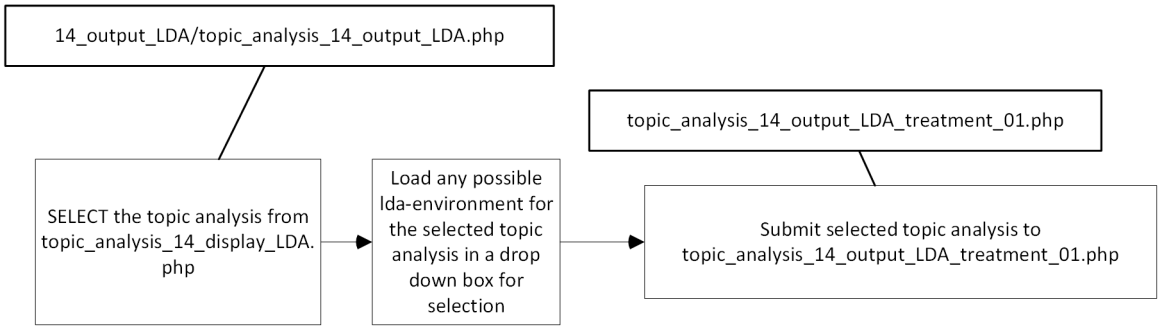


Figure 3.33 Flowchart: Selection of a latent dirichlet allocation environment which is related to the topic analysis environment in figure 3.32

identifier as a HTTP-POST-Variable. Another possibility is to access this output page with a HTTP-GET-Variable. This should enable a user to copy a URL of the output page of any existing latent dirichlet allocation environement to a webpage that is not part of this topic analysis tool. The output is therefore accessible by the topic analysis tool by the identifier as a http-post-variables and by external webpages by the identifier as a http-get-variable.
The output page consists of three information parts. The first information part is a summary of the selection that characterizes the selected latent dirichlet allocation environment and a link in order to be able to access this output from external webpages. The first information part will differ if either the output is loaded with the help of http-post-variable or a http-get-variable. The difference is illustrated in figure 4.75 and figure 4.76 in section 4.13 where the first mentioned figure shows the first information for a submitted latent dirichlet allocation environment identifier as a http-post-variable. The second mentioned figure shows the first information for the above mentioned identifier that is submitted as a http-get-variable. This reduced first information for the access from external webpages is a security arrangement in order to prevent the topic analysis tool from misusing by possible users that access the topic analysis tool from external webpages and try to use other functionalities which are not equal to this functionality.
Beside the first information part there is a second and third information part to show next. The second information part is the output of the latent dirichlet allocation and the third part is a listing of the scientific articles which provide the input text corpus for the executed latent dirichlet allocation. All three information

parts are introduced in section 4.13. The description of the topic analysis tool's implementation is finished by the flowchart in figure 3.34 which gives a survey of the circumstances that must be considered before the first, then the second and finally the third above mentioned information part is loaded.

3.4 Results

By implementing the topic analysis tool that is introduced between section 3.3.1 and section 3.3.11 we attempt to fulfill the requirements in section 3.1. A time-consuming task is to create features for the functionality "04_fill_and_complete" presented in section 3.3.4 that more or less solve the problem of fetching the right information from web search engines in order to save this information to the right place of the topic analysis tool where the topic analysis tool can find this information later again.
Time-consuming is to find the right web search engines for the missing values of scientific articles' attributes. If a web search engine is found that possibly provides the right information then this information must be extracted from search results of these web search engines after these web search engines are automatically requested with the help of the search string to return search results. This is because the most web search engines are not programmed for providing information for automatic tools like wget.exe. However, `http://dblp.org/search/publ/api` is neither complicated for requesting search results nor complicated for extracting returned search results to the database of the topic analysis tool because it is an API. `http://dblp.org/search/publ/api` is implemented as a filling-procedure as part of fill and complete for the "04_fill_and_complete" functionality.
Any other web search engine that already returns search results to the topic analysis tool's request is not API-based and therefore more complicated to handle because of their optimization for manual requests. Because of that reason the procedures that handle these type of web search engines like are more complicated. These procedures are already implemented as the completing procedures for "04_fill_and_complete" for requesting search results from `http://www.researchgate.net`. These completing procedures are shown as flowcharts in the appendix section D.
Functionalities for following preprocessing procedures and the functionality for

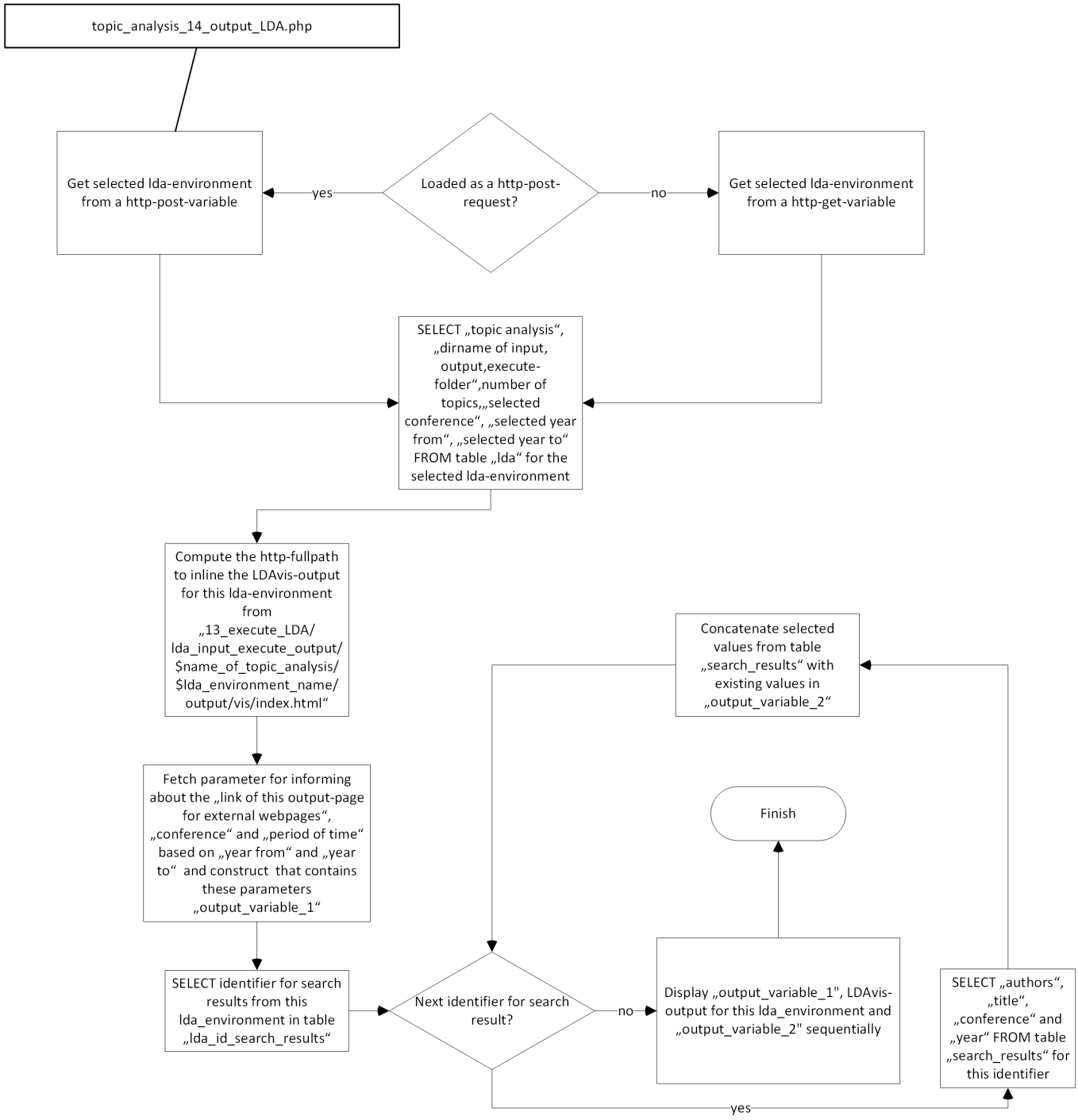


Figure 3.34 Flowchart: The output for the selected latent dirichlet allocation environment

preparing and displaying data for latent dirichlet allocation are easier to implement.

Chapter 4

Demonstration of the topic analysis tool for the category "Parsing" as a running example

The demonstration of a running example is for the category Parsing of the scientific field compiler building. We add "Parsing" to a topic analysis environment to the implemented topic analysis tool and show with the help of illustrations and descriptions how "Parsing" is transformed from an added topic analysis environment to an output of latent dirichlet allocation.
This transformation is possible because we visit each functionality shown in figure 4.1.

This demo can also be copied on the computer because this diploma thesis includes a DVD that contains the source code of the topic analysis tool. In the DVD's topic analysis tool's folder in "04_fill_and_complete/search_results/-Parsing" the search results from the web search engines `http://dblp.org/search/publ/api` and `http://www.researchgate.net` which are the basis for this demo can be found and used by the topic analysis tool. Any adjustments of the topic analysis tool are inside "00_general/php_functions.php". The SQL-file that is necessary to run the demo is saved in the root-folder of the "Demo" on the DVD.
We start with adding the category "Parsing" to a topic analysis environment in section 4.1.

Figure 4.1 The menu of the topic analysis tool

4.1 Adding "Parsing" as a new category

In this section we would like to add "Parsing" as a new category for a new topic analysis environment. In order to start with this plan we have to open the input field in "topic_analysis_01_add.php". What we see after entering "Parsing" is illustrated in figure 4.2.

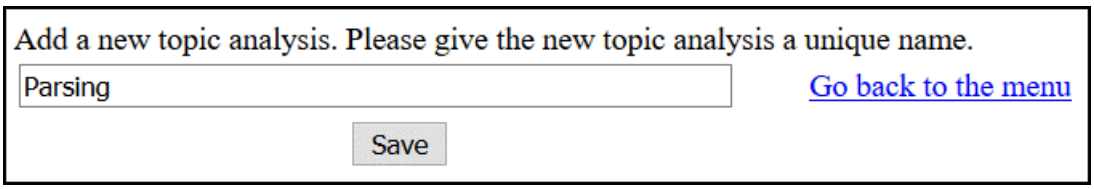


Figure 4.2 Enter "Parsing" as a new category's name

After submitting "topic_analysis_01_add.php" we see the text area field in figure 4.3 where we enter the research question for "Parsing".

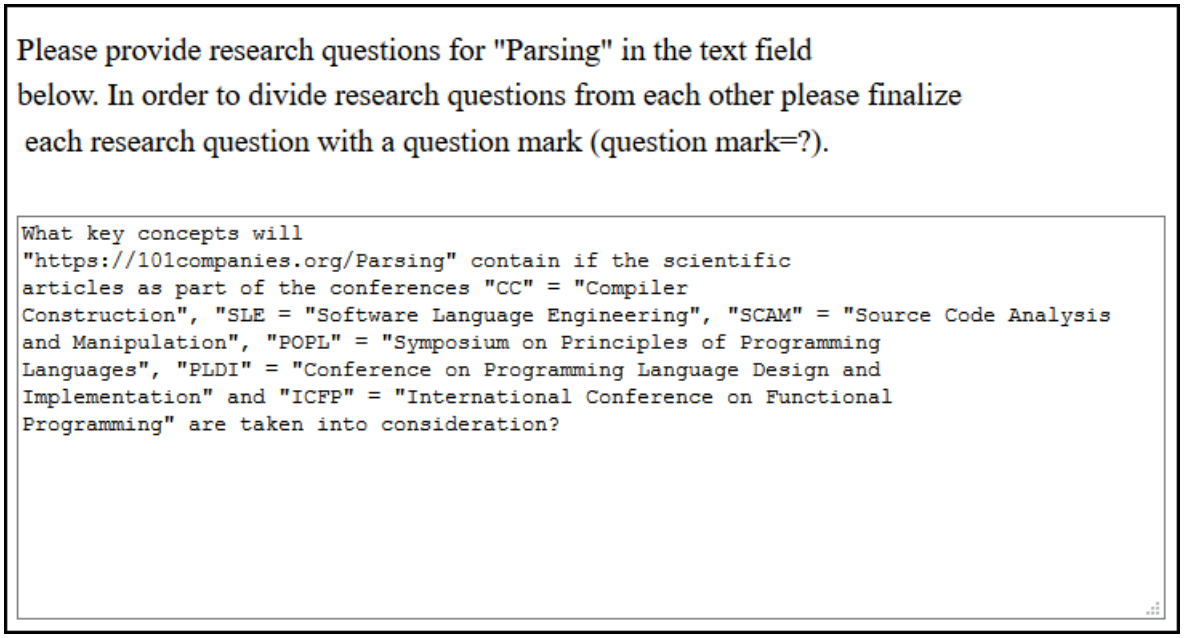


Figure 4.3 Enter research question(s) for "Parsing"

The next text area field is for the search strings for filling the topic analysis tool with search results. The search results are for "authors", "title", "year of publication" and "conference" of any scientific article that is a search result from http://dblp.org/search/publ/api. This search substring is the consequence of our research question from figure 4.3 above because we wanted scientific articles from several conferences that are fetchable as search results from http://dblp.org as illustrated in figure 4.4.

"Abstracttext" and "fulltext" for the fetched search results from http://dblp.org must be from http://www.researchgate.net. Therefore, we complete the already saved search results for "authors", "title", "year of publication" and "conference" with search results from http://www.researchgate.net for "abstracttext" and "fulltext" by entering inputs for the next text area field in figure 4.5. However, we should read the tool's instructions how to complete. These provided instructions are illustrated in figure 4.6 and in figure 4.7.

After the inputs are entered as illustrated above we submit these entries by activating the button with the value "Save". Then the category "Parsing" is added and can be used for collecting search results to the topic analysis tool.

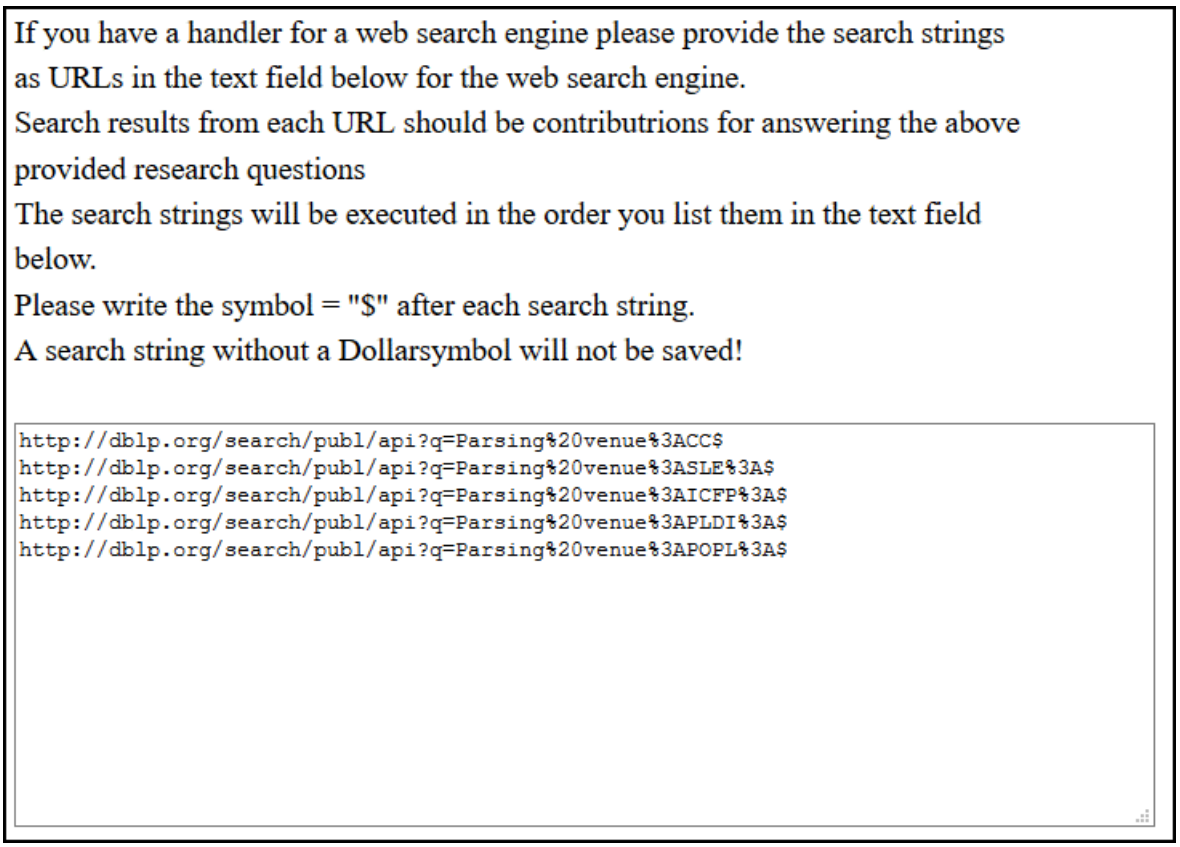


Figure 4.4 Enter search strings for "Parsing"

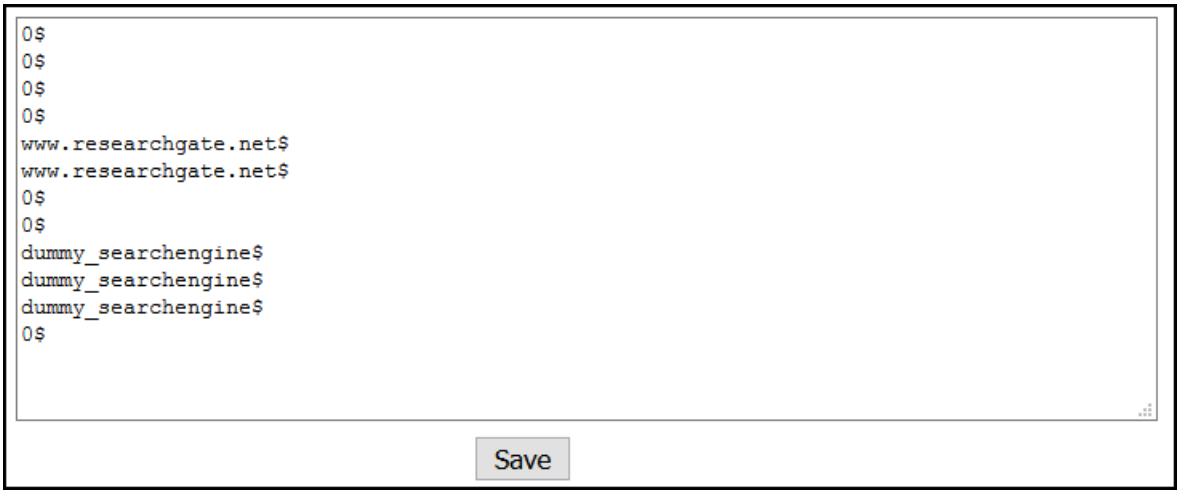


Figure 4.5 Search strings for `http://www.researchgate.net` according to the notice from below

The picture beside illustrates how rows of the text area field below the picture have to be filled.
If you have a handler to complete the search then write this handler's identifier in the i-th row where the i-th column is an attribute in the picture beside. If you do not have a handler for this attribute please write 0$ in this related row instead. Any row must be finished by a Dollarsymbol "$" otherwise the parameter for a row will not be saved.
The following handler for the following attributes already exist:
www.researchgate.net$ for "first_link_to_abstracttext",
www.researchgate.net$ for "abstracttext",
dummy_searchengine$ for "path_to_pdffulltext",
dummy_searchengine$ for "pdffulltext_as_text" and
dummy_searchengine$ for "pdffulltext_as_text_extracted.

Figure 4.6 What has to be noticed for the search strings for `http://www.researchgate.net`

4.2 Modifying "Parsing" as an existing category

Because we already added "Parsing" we optionally can modify the name for "Parsing", the research questions or the search strings. Modifying the research questions and the search strings is identical to adding research questions and the search strings as already shown in section 4.1. The only difference to the input possibilities presented in section 4.1 is that "Parsing" is not added but selected as a category in "topic_analysis_02_modify.php" as illustrated in figure 4.8.

After submitting "topic_analysis_02_modify.php" we can change the name for the topic analysis environment's name which is "Parsing" as illustrated in figure 4.9.

Modifying research questions and search strings is already shown in section 4.1. Any modification is saved by activating the button with the value "Save". Then Then the category "Parsing" is modified and can be used for collecting search results to the topic analysis tool which is presented as a functionality after having shown the deletion of a topic analysis environment.

4.3 Deleting "Parsing" as an existing category

If "Parsing" is not longer needed then any database information and information in folder and files that contain the collected search results can be deleted.

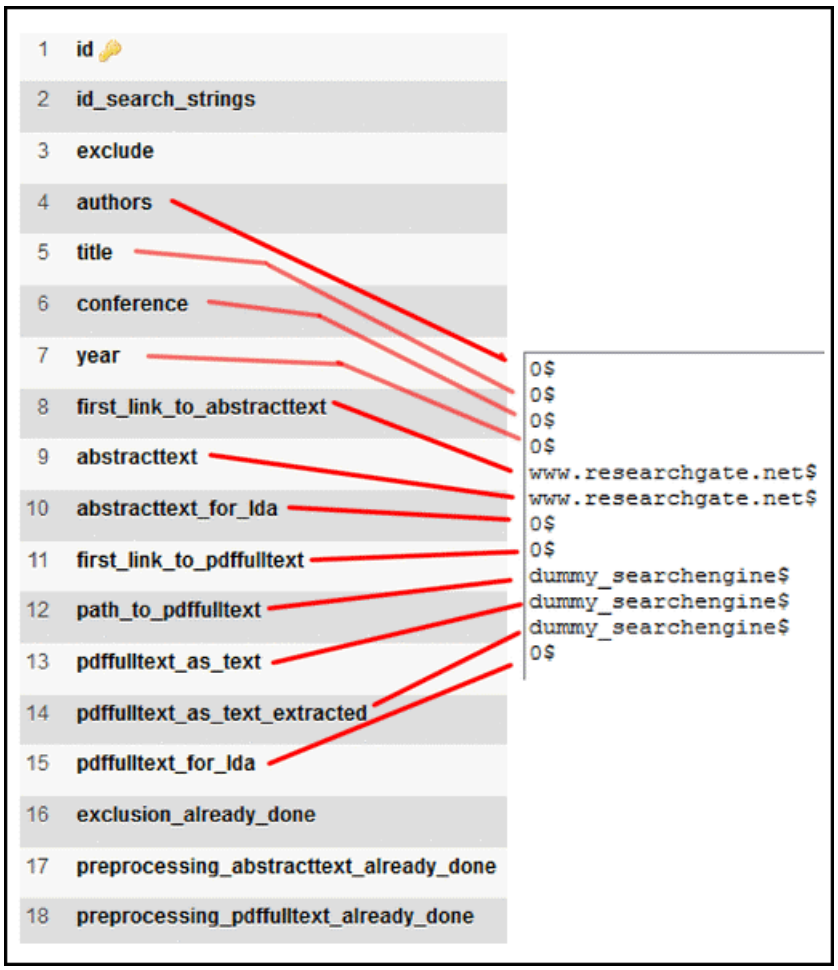


Figure 4.7 Visualization for the notice from above

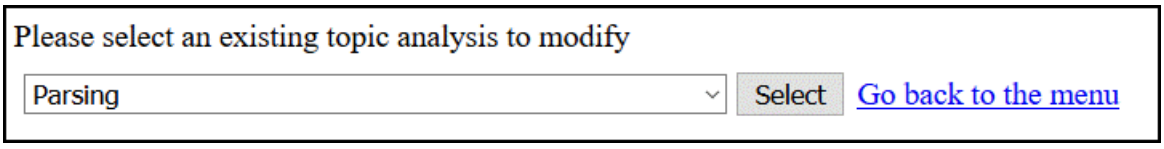


Figure 4.8 Select "Parsing" as an existing category's name

To begin with the deletion procedure the start page for the deletion of any above mentioned information in "topic_analysis_03_delete.php" is loaded.
This start page provides a drop-down field where any possible topic analysis environment can be selected. If "Parsing" as shown in figure 4.12 is selected and submitted then the confirmation page as shown in figure 4.13 is loaded and dis-

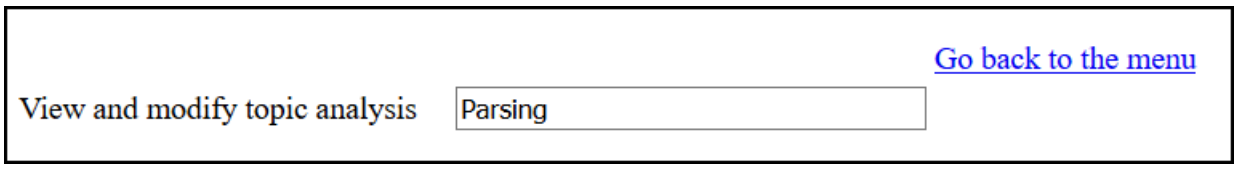


Figure 4.9 Input field for modifying "Parsing" as the existing topic analysis environment's name

played in order that the user decides to confirm or not to confirm that the topic analysis should be deleted irrevocably.

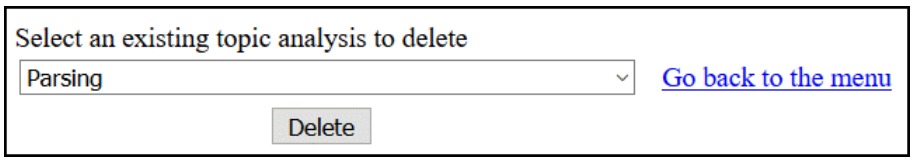


Figure 4.10 Flowchart for selecting a topic analysis environment's name

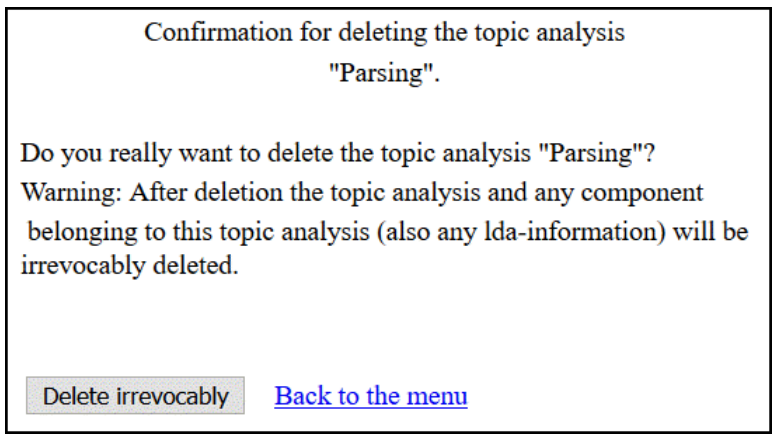


Figure 4.11 Flowchart for confirming the deletion of a topic analysis environment

After having decided to delete "Parsing" any information in files and database tables concerning the topic analysis environment and the environment for latent dirichlet allocation is deleted.

4.4 Collect search results for "Parsing"

With the help of this functionality it is possible to collect the search results from web search engines that are known to the topic analysis environment[1]. If this is the case then we can submit "Parsing" which is illustrated in figure 4.12.

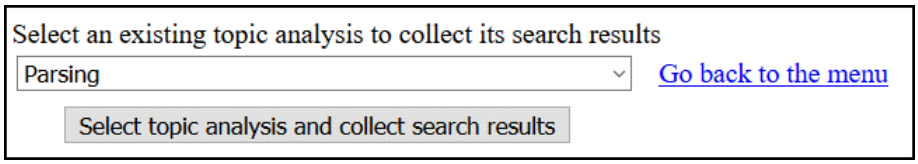


Figure 4.12 User input of "Parsing" in "topic_analysis_04_fill_and_complete_search_results.php"

After the topic analysis environment for "Parsing" is submitted the topic analysis tool selects any search string in figure 4.13 one after another in order to fetch the search result from the web search engine that is addressed by the selected search string.

The topic analysis tool will return the message in figure 4.14 if the search results are fetched successfully. According to this message we also can look into the search result.

Up to this point we filled the attributes "Authors", "Title", "Year" and "Conference". We can also look into the snapshot of the place where the topic analysis tool saved the search results for these attributes. Figure 4.15 is an excerpt of this snapshot.

Also visible in Figure 4.15 is the submit button for completing search results. This is a notice that the scientific articles we partly see in the snapshot are not complete. These scientific articles need the information for abstracttext and fulltext. For that reason the button "Complete search results" must be activated next. The first what happens after the button is activated is that the link in figure 4.16 is shown.

If this link is activated then the links to abstracttexts are downloaded in html-files from `http://www.researchgate.net`. However, these links are hidden in the search results. Because of this circumstance the links are extracted from the search results from `http://www.researchgate.net` and saved in the attribute "first_link_to_abstracttext" for each affected search result in the topic analysis tool. The result is shown in the snapshot in figure 4.17.

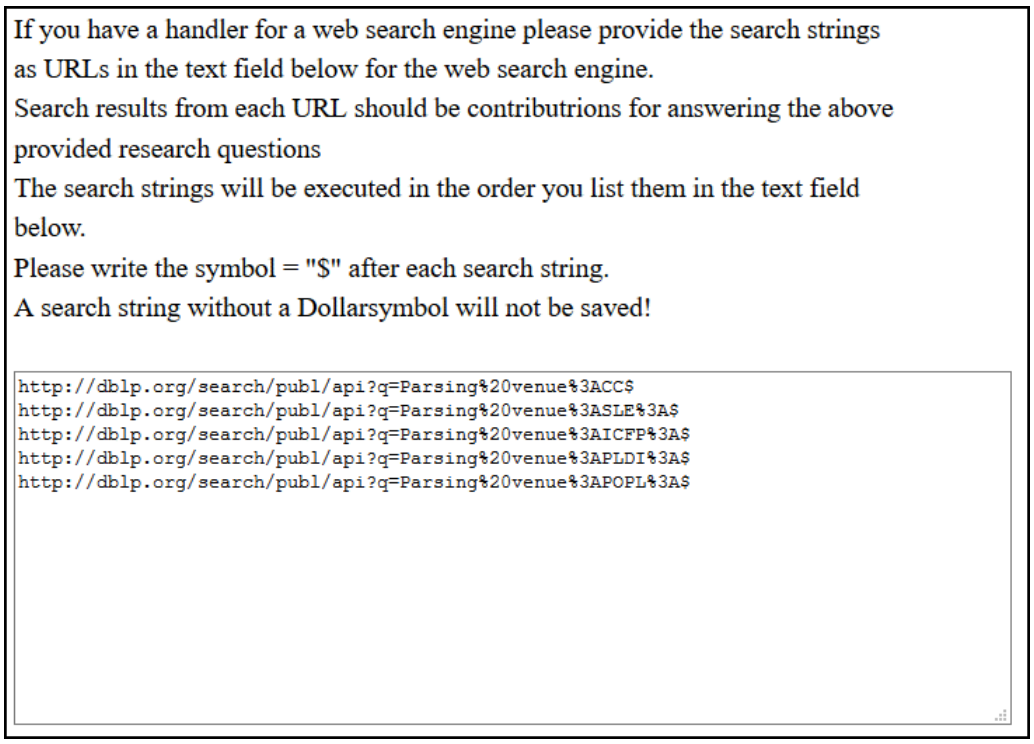


Figure 4.13 Search strings we enter in the second field in the form for adding a topic analysis environment

With the help of these links the abstracttexts are fetched next. We will achieve this if we activate the link in figure 4.18.

If we activate this link the attribute "abstracttext" is filled with extracted abstracttexts from the search results of `http://www.researchgate.net` as it is illustrated in figure 4.19.

While extracting abstracttexts from the search results of `http://www.researchgate.net` parts of some search result files contain the link to the fulltext beside the abstracttext that is also available from `http://www.researchgate.net`. We will extract these links to fulltexts because we must use the possibility to overcome the paywall of some web search engines because of requirement 5 in section 3.1. The result is that we have a "first_link_to_pdffulltext" for some search results in the topic analysis tool as it is illustrated in figure 4.20.

Execution of topic analysis "Parsing".
The 1. run was successful.
The 2. run was successful.
The 3. run was successful.
The 4. run was successful.
The 5. run was successful.
Please have a look into the search result of each search string.
If any search result does not satisfy your needs please modify this search string to your greatest advantage.

Search result for the 1. search string for verification purpose.
Search result for the 2. search string for verification purpose.
Search result for the 3. search string for verification purpose.
Search result for the 4. search string for verification purpose.
Search result for the 5. search string for verification purpose.

Figure 4.14 Message of the topic analysis tool to inform that the collection of search results is successful

The information in table "search_results" from the search engine because of your search strings for this topic analysis are in the table below. If anything is in your favour please complete the search results in line of your entries in the third text area of the html-form for modifying components for this topic analysis.

Complete search results

Number	id	id_search_strings	exclude	authors	title	conference	year	first_link_to_abstracttext
1	271	862	0	Ali Afroozeh, Anastasia Izmaylova	Iguana - a practical data-dependent parsing framework.	CC	2016	

Figure 4.15 Excerpt of the place where the topic analysis tool saves the search results

Figure 4.16 Link to download and extract "links to abstracttexts" from www.researchgate.net

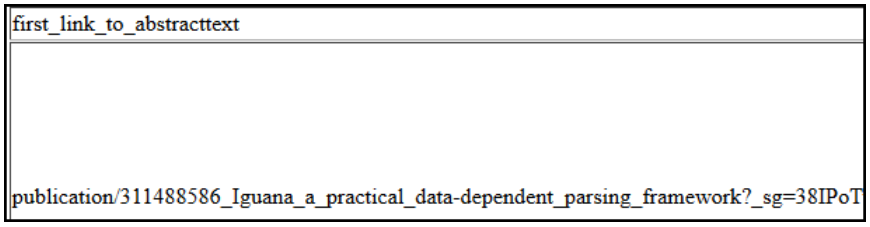


Figure 4.17 Added links to abstracttexts to the topic analysis tool

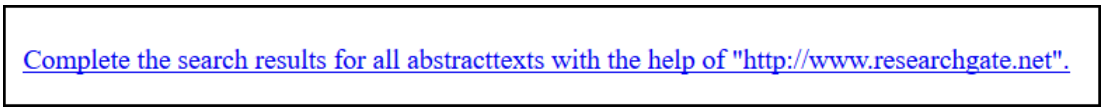


Figure 4.18 Link to download and extract abstracttexts from www.researchgate.net

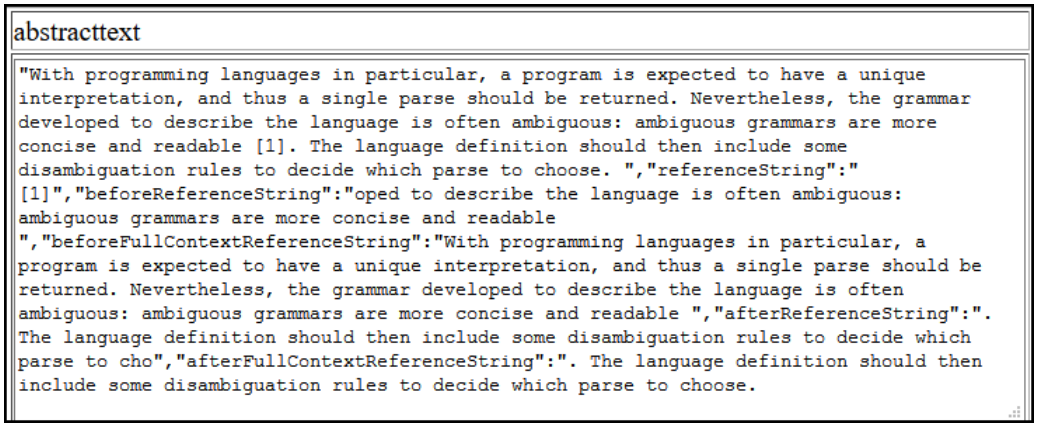


Figure 4.19 Added abstracttexts to the topic analysis tool

In the next step we read each value of "first_link_to_pdffulltext", download the related fulltexts and save the fullpath of these related and downloaded fulltexts to "path_to_pdffulltext". In order to fullfil this step we activate the link that is

[1]The topic analysis tool checks in this functionality whether procedures exist that correctly transfer the search results from the requested web search engine to the database of the topic analysis tool.

Figure 4.20 An added link to "first_link_to_pdffulltext" to the topic analysis tool

shown after the values are completely saved in "first_link_to_pdffulltext" and that is illustrated in figure 4.21.

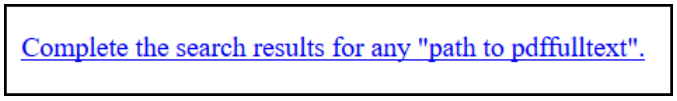


Figure 4.21 Link to download fulltexts for "first_link_to_pdffulltext"

If this link to download the fulltexts according to the values of "first_link_to_pdffulltext" is activated then the fulltexts are downloaded and for each downloaded fulltext the fullpath to the downloaded fulltext is saved as a value in "path_to_pdffulltext". Figure 4.22 extracted from the snapshot shows a relation between a value of "first_link_to_pdffulltext" and a value of "path_to_pdffulltext".

first_link_to_pdffulltext	path_to_pdffulltext
https://www.researchgate.net/profile/Baobao_Chang2/publication/308960326_Improved_Graph-Based_Dependency_Parsing_via_Hierarchical_LSTM_Networks/links/59293f0baca27295a806ff7c/Improved-Graph-Based-Dependency-Parsing-via-Hierarchical-LSTM-Networks.pdf	C:/xampp/htdocs /topic_analysis /04_fill_and_complete /search_results /Parsing/pdffulltext /all/Improved-Graph-Based-Dependency-Parsing-via-Hierarchical-LSTM-Networks.pdf

Figure 4.22 Relation between a link in "first_link_to_pdffulltext" and a fullpath in "path_to_pdffulltext"

In order to be able to preprocess text of each downloaded fulltext in a later functionality we extract the text of each downloaded fulltext. We have to activate the link illustrated in figure 4.23 to start this procedure.

Complete the search results for the conversion of any pdffulltext to text.

Figure 4.23 Link to extract text from the downloaded fulltexts

This procedure extracts the text in a first step and save the extracted text as a value for "pdffulltext_as_text". The result is shown from an excerpt of the snapshot in figure 4.24 where the left value belongs to "first_link_to_pdffulltext", the value in the middle to "path_to_pdffulltext" and the left value to "pdffulltext_as_text".

If we follow the last link in terms of completing the search results illustrated in figure 4.25 we give the user the possibility to manually modify each text saved in "pdffulltext_as_text" one step before.

https://www.researchgate.net/profile/Baobao_Chang2/publication/308960326_Improved_Graph-Based_Dependency_Parsing_via_Hierarchical_LSTM_Networks/links/59293f0baca27295a806ff7c/Improved-Graph-Based-Dependency-Parsing-via-Hierarchical-LSTM-Networks.pdf	C:/xampp/htdocs /topic_analysis /04_fill_and_complete /search_results /Parsing/pdffulltext /all/Improved-Graph-Based-Dependency-Parsing-via-Hierarchical-LSTM-Networks.pdf	As we can see, word representations combining character-level information do improve model's performance compared with [18]. Moreover, the LAS of our model achieves state-of-the-art accuracy. Table 1. Comparison with previous state-of-the-art models on CTB5. Method CTB5 UAS LAS Zhang and Nivre [23] 86.0 84.4 Bernd Bohnet [6] 87.5 85.9 Zhang and McDonald [22] 87.96 86.34 Dyer et al. [10] 87.2 85.7 Ballesteros et al. [5] 85.30

Figure 4.24 Saved extracted text for the right column "pdffulltext_as_text"

Show the converted pdffulltexts for the manual extraction.

Figure 4.25 Link for the manual modification of texts in "pdffulltext_as_text"

It is mandatory to submit the manual modification page for the texts from "pdffulltext_as_text". However, it is not mandatory to modify any text on this page. If no modification to any text on this page is submitted the procedure need not to be repeated again. Figure 4.26 shows an excerpt of this modification page with the submit-button at the end of the modification page.
After submitting the modification page the texts from "pdffulltext_as_text" are moved to "pdffulltext_as_text_extracted" as modfied or unmodified extracted texts.
If we are finished with completing any scientific articles' attribute with values then the execution of this functionality is finished. The message in figure 4.27 is a notice for this circumstance.
The notice in figure 4.27 is a preparation for the next functionality because we need more fulltexts for latent dirichlet allocation which is prepared in the functionality before the last functionality of this topic analysis tool.

4.5 Manual upload of fulltexts for "Parsing"

Because fill and complete is finished with the help of the previous functionality and because we need more fulltexts for the analysis part of this topic analysis tool which is fulfilled by latent dirichlet allocation[2] we upload more fulltexts we

[2]See section 2.2 for information on latent dirichlet allocation.

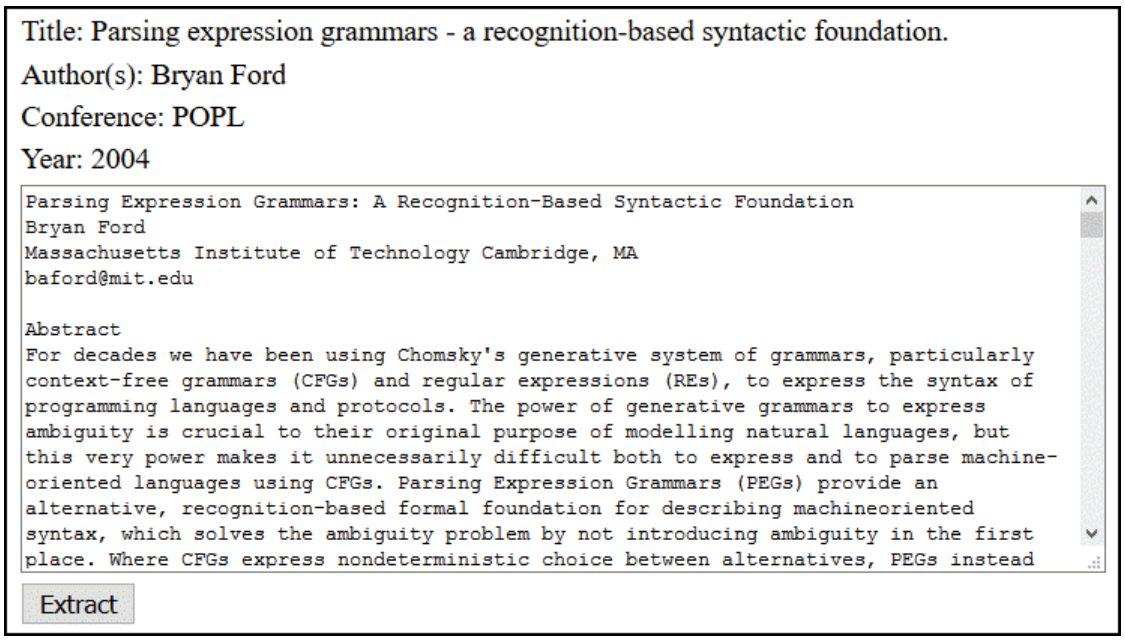


Figure 4.26 Extracting texts from "pdffulltext_as_text" to "pdffulltext_as_text_extracted" after submit

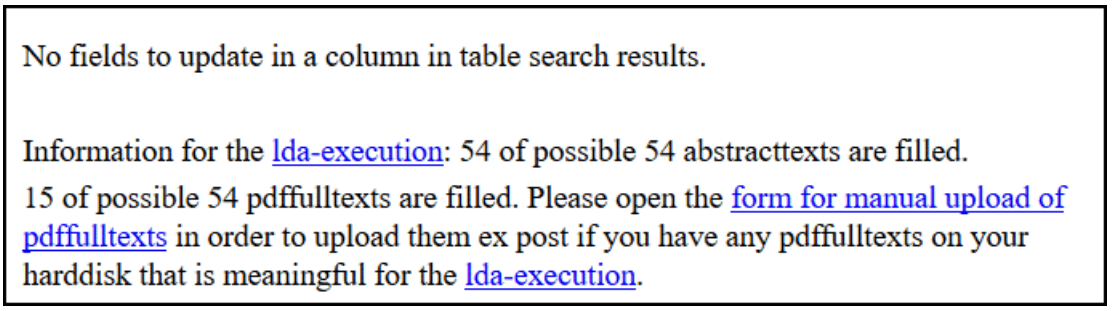


Figure 4.27 Notice that the execution of this functionality is finished

already downloaded from web search engines[3]. This is easier because a lot of web search engines for scientific articles prevent automatic tools like wget.exe from downloading fulltexts from their platform areas that only can be accessed by their subscribers.
The order of uploading a manually downloaded fulltext for "Parsing" is as follows: We start by selecting "Parsing" as illustrated in figure 4.28.

[3]This is a possibility to overcome the paywall of particular web search engines based on requirement 5 and requirement 6 in section 3.1.

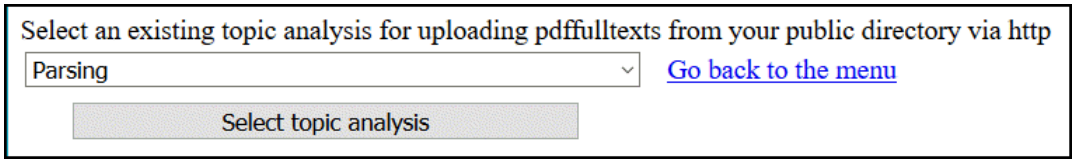


Figure 4.28 User input of "Parsing" in "topic_analysis_05_upload_further_pdffulltexts.php"

After having submitted "Parsing" the upload page for additional fulltexts is loaded and displayed. This page consists of a snapshot. We navigate to the column with the headline "path_to_pdffulltext" and look for a field that has an input field instead of a fullpath for "path_to_pdffulltext". An input field for "path_to_pdffulltext" means that a fulltext has not been uploaded for the scientific article of this row yet. An example for this input field is illustrated in figure 4.29.

path_to_pdffulltext	pdffulltext_as_text	pdffulltext_as_text_extracted
C:/xampp/htdocs/topic_analysis/04_fill_and_complete /search_results/Parsing/pdffulltext/all/Improved-Graph-Based-Dependency-Parsing-via-Hierarchical-LSTM-Networks.pdf	This text was deleted because it was replaced by the manually extracted text in column pdffulltext_as_text_extracted.	Improved Graph-Based Dependen Wenhui Wang1,2 and Baobao Cha 1 Key Laboratory of Computati Electronics Engineering and C Peking University, No. 5 Yihe {wangwenhui,chbb}@pku.edu.cn 2 Collaborative Innovation Ce Abstract. In this paper, we p which utilizes hierarchical L word representations, allowin and capture both distribution achieves state-ofthe-art accu English Penn Treebank with on effectiveness in recovering d Keywords: Graph-based depende

Figure 4.29 Snapshot that shows that a fulltext for the first scientific article has not been uploaded yet

Thus we scroll back to the beginning of the row and look for the title of the scientific article. If we have access we download the fulltext from a web search engine in order to upload this fulltext to a public directory in the internet that can be accessed by the topic analysis tool in the next step. In this next step we enter the http-fullpath to the empty input field in figure 4.29 as illustrated in figure 4.30.

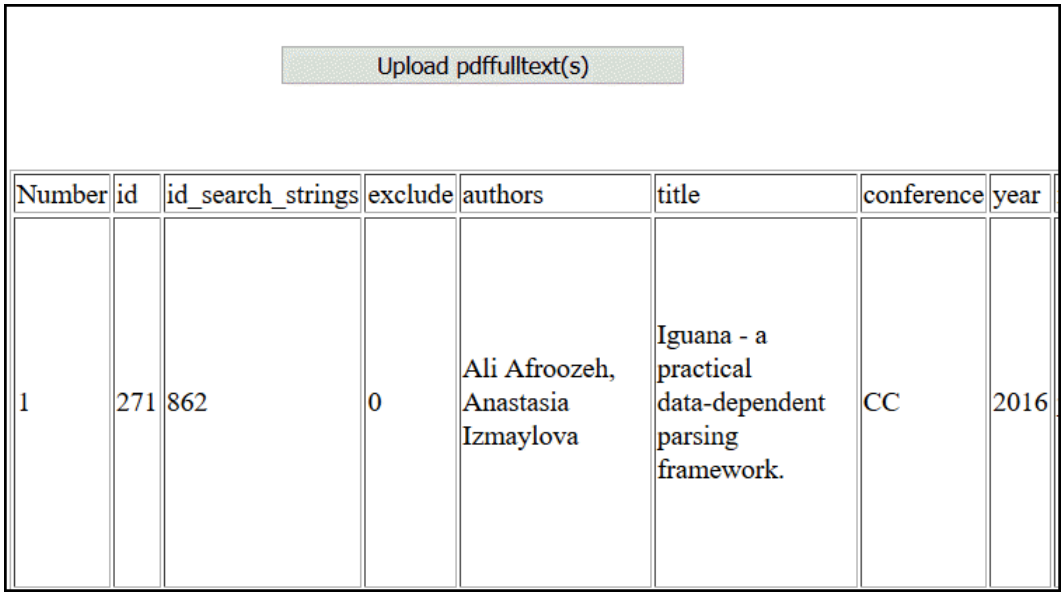
Figure 4.30 A filled input field with a http-fullpath to the fulltext

The fulltext must be accessed with the help of the http-protocol because we need an unauthorized access to the fulltext in this public directory which cannot be always taken for granted if the ftp-protocol is used.
If we submit the upload of this fulltext e.g. from `https://userpages.uni-koblenz.de/~tschnitzler/p267-afroozeh.pdf` with the help of the submit-button illustrated in figure 4.31 above any information concerning this scientific article then the fulltext will be uploaded to the topic analysis tool.

After having activated the submit button the input field as it is shown in figure 4.29 disappears on the page after the submit. Instead of the input field the local fullpath to this fulltext is shown. Moreover, the fulltext is converted to text. This text is saved as a field value for "pdffulltext_as_text_extracted". Any difference on the page after the submit is shown in figure 4.32.

4.6 Exclude search results for "Parsing"

This functionality is for excluding the scientific articles that are not relevant for studying and summarizing "Parsing" that we get from the "filling-procedure" in



Upload pdffulltext(s)

Number	id	id_search_strings	exclude	authors	title	conference	year
1	271	862	0	Ali Afroozeh, Anastasia Izmaylova	Iguana - a practical data-dependent parsing framework.	CC	2016

Figure 4.31 The submit button that is at the top of the scientific article the fulltext is uploaded for

path_to_pdffulltext	pdffulltext_as_text	pdffulltext_as_text_extracted
C:/xampp/htdocs/topic_analysis/04_fill_and_complete/search_results/Parsing/pdffulltext/all/p267-afroozeh.pdf	This text was deleted because it was replaced by the manually extracted text in column pdffulltext_as_text_extracted.	Iguana: A Practical Data-Dependent Parsing Framework Ali Afroozeh Anastasia Izmaylova Centrum Wiskunde & Informatica, Amsterdam, The Netherlands {ali.afroozeh, anastasia.izmaylova}@cwi.nl Abstract Data-dependent grammars extend context-free grammars with arbitrary computation, variable binding, and constraints. These features provide the user with the freedom and power to express syntactic constructs outside the realm of context-free grammars, e.g., indentation rules in Haskell and type definitions in C. Data-dependent grammars have been recently presented by Jim et al. as a grammar formalism that enables construction of parsers from a rich format specification. Although some features of data-dependent grammars are available in current parsing tools, e.g., semantic predicates in ANTLR, data-dependent grammars have not yet fully found their way into practice.

Figure 4.32 Changes take effect to fields in "path_to_pdffulltext" and "pdffulltext_as_text_extracted"

terms of fill and complete because the filling-procedure uses search strings we formulate as part of the research questions according to [5, slide 12-16].
In order to start with the exclusion-procedure we select "Parsing" as illustrated in figure 4.33.

After having submitted "Parsing" the exclusion page for is loaded and displayed. This page consists of write-protected research questions for "Parsing" and a snapshot where the scientific articles have to be excluded if they do not answer the displayed research questions according to [5, slide 12-16]. The research question

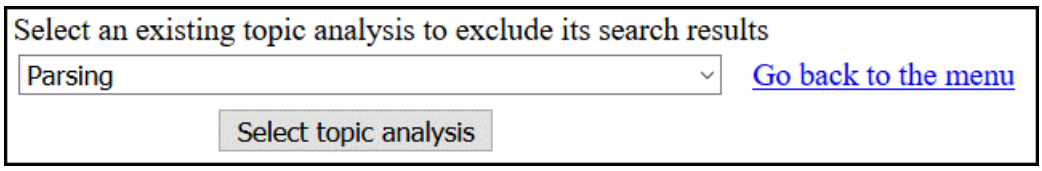


Figure 4.33 User input of "Parsing" in "topic_analysis_06_exclude_search_results.php"

is shown in figure 4.34. An excerpt of the snapshot with the search results to be possible to exclude is shown in figure 4.35.

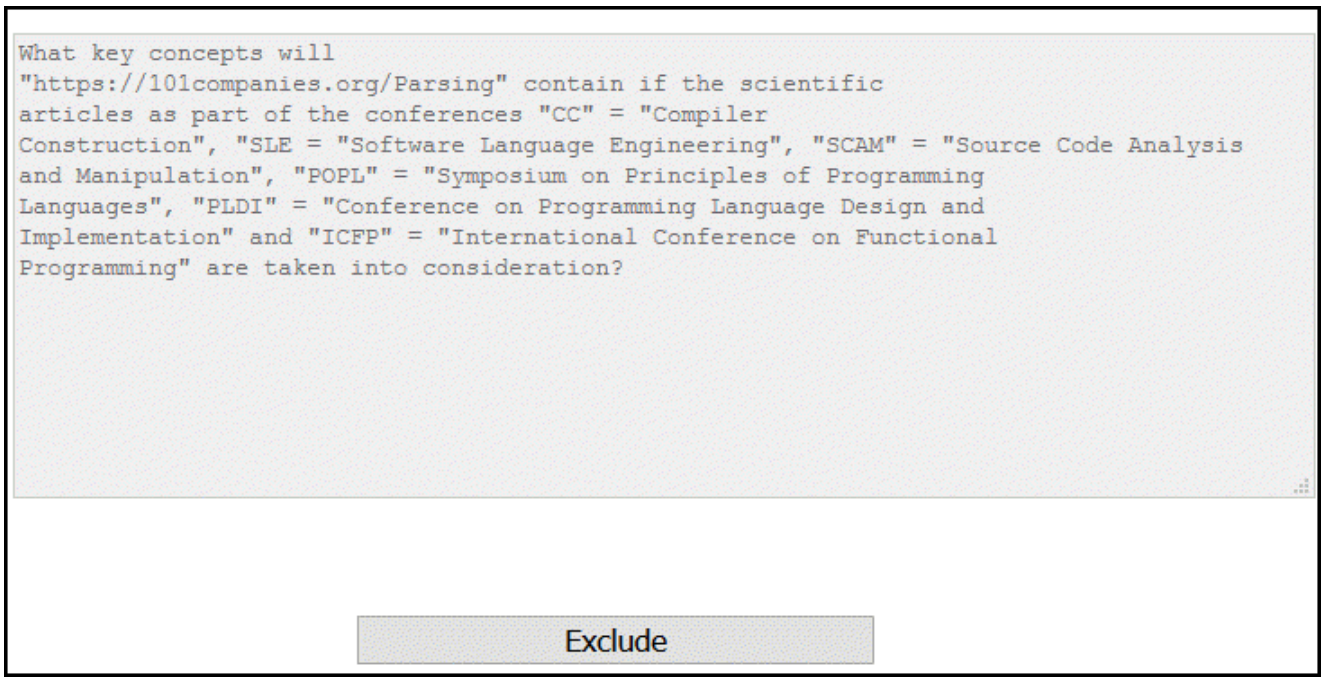


Figure 4.34 Write-protected research questions for reminding the research questions for "Parsing" and the exclude-button for submitting

If a scientific article is checked then it will be excluded for further functionalities. An already checked scientific article can be unchecked again later in order to include it further functionalities again.

4.7 Preprocess abstracts for "Parsing"

After the exclusion procedure executed with the help of the previous functionality the preprocessing procedure for abstracttexts starts. The preprocessing is necessary for the execution of latent dirichlet allocation that is prepared in functionality

Number	id	id_search_strings	exclude	authors	title	conference	year
1	2534	882	☐ exclude	Ali Afroozeh, Anastasia Izmaylova	Iguana - a practical data-dependent parsing framework.	CC	2016
2	2535	882	☐ exclude	Wenhui Wang, Baobao Chang	Improved Graph-Based Dependency Parsing via Hierarchical LSTM Networks.	CCL	2016

Figure 4.35 A snapshot's excerpt of the search results that can be checked in terms of exclusion

"topic_analysis_13_execute_LDA.php". In order to optimize the execution of latent dirichlet allocation the input for this execution must consist of words from a natural language like "english" or "german".
For that reason the procedure for preprocessing excludes text characters that are not an extension to understand sentences based on alphabetic characters.
In order to start with preprocessing abstracttexts we select "Parsing" as illustrated in figure 4.36.

Next "Parsing" must be submitted by activating the button with the value "Select topic analysis". After submitting the preprocessing procedure for abstracttexts this procedure runs automatically for any abstracttext of this topic analysis environment until no abstracttext is left for preprocessing. After preprocessing a snapshot is shown in order to be informed about the difference between input

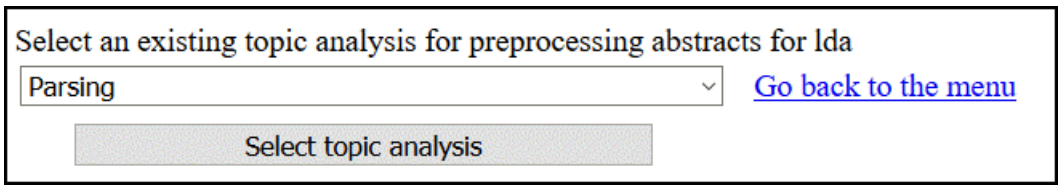


Figure 4.36 User input of "Parsing" in "topic_analysis_07_a_preprocessing_abstracts.php"

and output of preprocessing. An excerpt of the snapshot from figure 4.37 to figure 4.38 shows from left to right the authors, the title, the year, the conference and the first link to abstracttext of the scientific articles that are preprocessed.

Preprocessing for pdffulltexts_as_text_extracted for topic analysis "Parsing". Back to the

The preprocessing results from column "abstracttext" are saved in the column "abstracttext_for_lda".

Number	id	id_search_strings	exclude	authors	title	conference	year	first_
1	271	862	0	Ali Afroozeh, Anastasia Izmaylova	Iguana - a practical data-dependent parsing framework.	CC	2016	publi

Figure 4.37 A snapshot-excerpt of authors, title, year, conference and ...

To be able to see the difference between the input and the output of the preprocessing procedure for abstracttexts for "Parsing" the snapshot provides the abstracttexts in column "abstracttext" for representing the preprocessing input and the abstracttexts in column "abstracttext_for_lda" which is the preprocessing's output. The difference between input and output in these two columns of the snapshot is illustrated in figure 4.39 and in in figure 4.40.

As it is already explained above it is recommended to exclude words from the preprocessed abstracttexts that are not part of a natural language. However, preprocessed abstracttexts can still consist of words that are not a contribution for studying and summarizing "Parsing". For that reason this topic analysis tool's functionality "topic_analysis_08_optimize_abstracts.php" should be used in order to delete superfluous parts in abstracttexts.

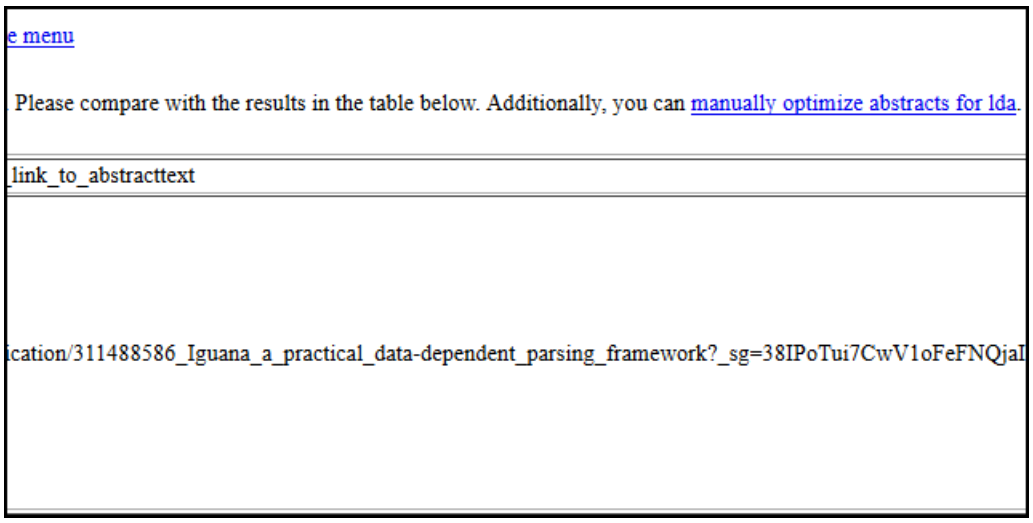


Figure 4.38 ... first link to abstracttext of affected scientific articles that are preprocessed

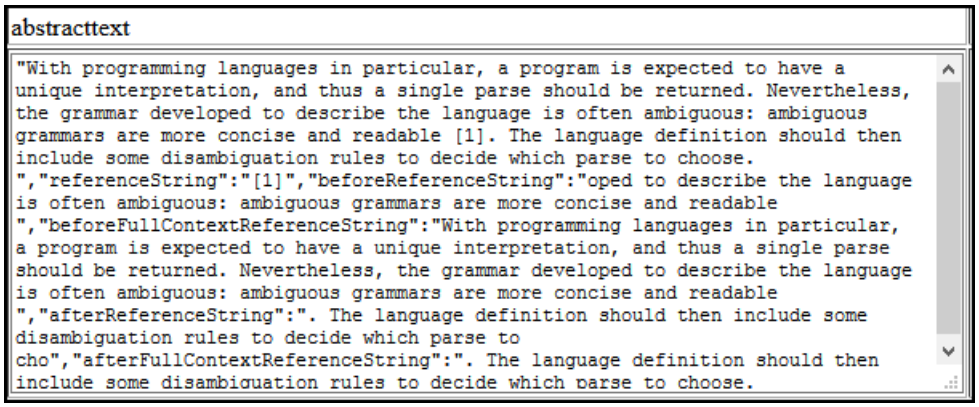
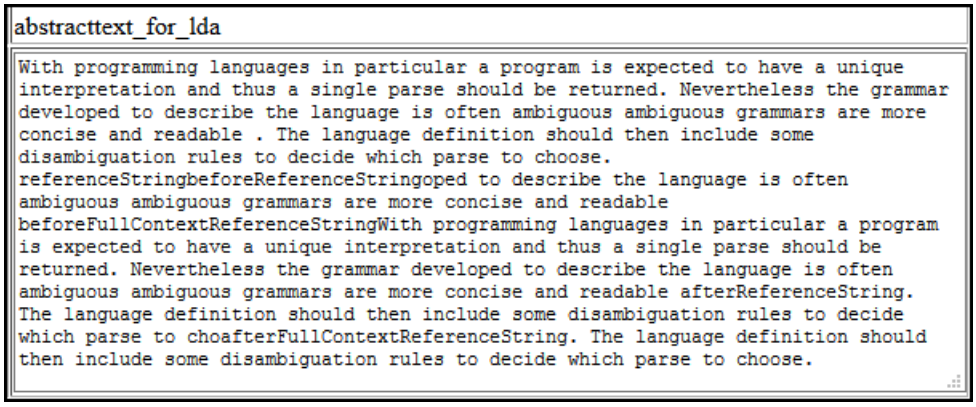


Figure 4.39 An excerpt of the input for preprocessing in the snapshot

4.8 Preprocess fulltexts for "Parsing"

As already noted preprocessing is necessary for the abstracttexts and fulltexts that are the inputs for the execution of latent dirichlet allocation in line of this topic analysis tool.
The preprocessing for fulltexts follows the principle presented in section 4.7 for preprocessing abstracttexts.
In order to start with the procedure for preprocessing fulltexts we select "Parsing" as illustrated in figure 4.41.



abstracttext_for_lda
With programming languages in particular a program is expected to have a unique interpretation and thus a single parse should be returned. Nevertheless the grammar developed to describe the language is often ambiguous ambiguous grammars are more concise and readable . The language definition should then include some disambiguation rules to decide which parse to choose. referenceStringbeforeReferenceStringoped to describe the language is often ambiguous ambiguous grammars are more concise and readable beforeFullContextReferenceStringWith programming languages in particular a program is expected to have a unique interpretation and thus a single parse should be returned. Nevertheless the grammar developed to describe the language is often ambiguous ambiguous grammars are more concise and readable afterReferenceString. The language definition should then include some disambiguation rules to decide which parse to choafterFullContextReferenceString. The language definition should then include some disambiguation rules to decide which parse to choose.

Figure 4.40 An excerpt of the preprocessing's output in the snapshot

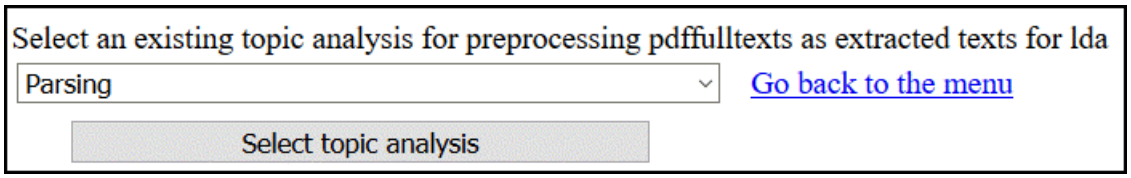


Figure 4.41 User input of "Parsing" in "topic_analysis_07_b_preprocessing_pdffulltexts.php"

The preprocessing of fulltexts leads to the information in the excerpt of the snapshot as illustrated in figure 4.42 and in figure 4.43 from left to right.

Preprocessing for pdffulltexts_as_text_extracted for topic analysis "Parsing". Back to the menu

The preprocessing results from column "pdffulltexts_as_texts_extracted" are saved in the column "pdffulltext_for_lda".

Number	id	id_search_strings	exclude	authors	title	conference	year	first_link_to_abstracttex
1	271	862	0	Ali Afroozeh, Anastasia Izmaylova	Iguana - a practical data-dependent parsing framework.	CC	2016	publication/311488586_

Figure 4.42 Left part of the information concerning addtional optimization of fulltexts

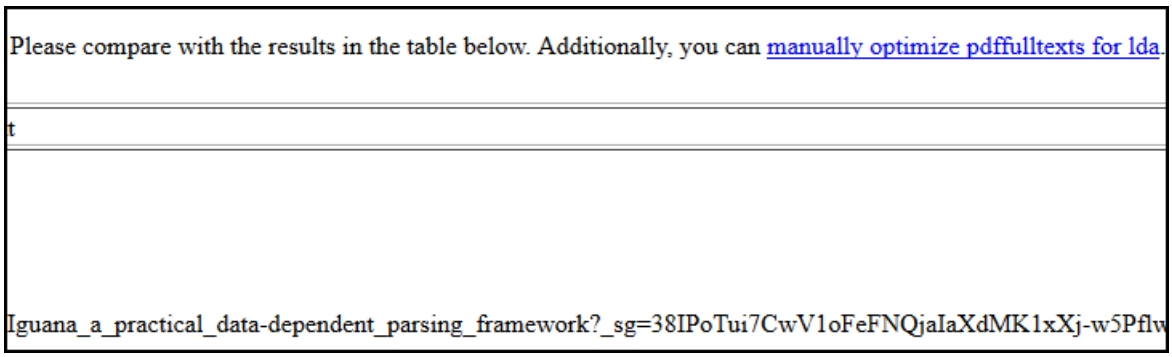


Figure 4.43 Right part of the information concerning addtional optimization of fulltexts

To be able to see the difference between the input and the output of the preprocessing procedure for fulltexts for "Parsing" the snapshot provides the fulltexts in column "pdffulltext_as_text_extracted" for representing the preprocessing input and the fulltexts in column "pdffulltext_for_lda" which is the preprocessing's output. The difference between input and output in these two columns of the snapshot is illustrated in figure 4.44 and in in figure 4.45.

As it is already explained above it is recommended to exclude words from the preprocessed fulltexts that are not part of a natural language. However, preprocessed fulltexts can still consist of words that are not a contribution for studying and summarizing "Parsing". For that reason this topic analysis tool's functionality "topic_analysis_09_optimize_pdffulltexts_for_lda.php" should be used in order to delete superfluous parts in fulltexts.

4.9 Optimize abstracts for "Parsing"

Abstracttexts should be optimized after preprocessing because the output of preprocessing can contain words that are not in the set of words of a selected natural language. We want the user to delete superfluous parts of abstracttexts that are not a contribution for studying and summarizing "Parsing". In order to start with this functionality we select "Parsing" as illustrated in figure 4.46.

After submitting "Parsing" the preprocessed abstracttexts are loaded. In order to refer to the problem that some words in abstracts or fulltexts cannot be related to a natural language we focus on an loaded abstracttext that contains at least

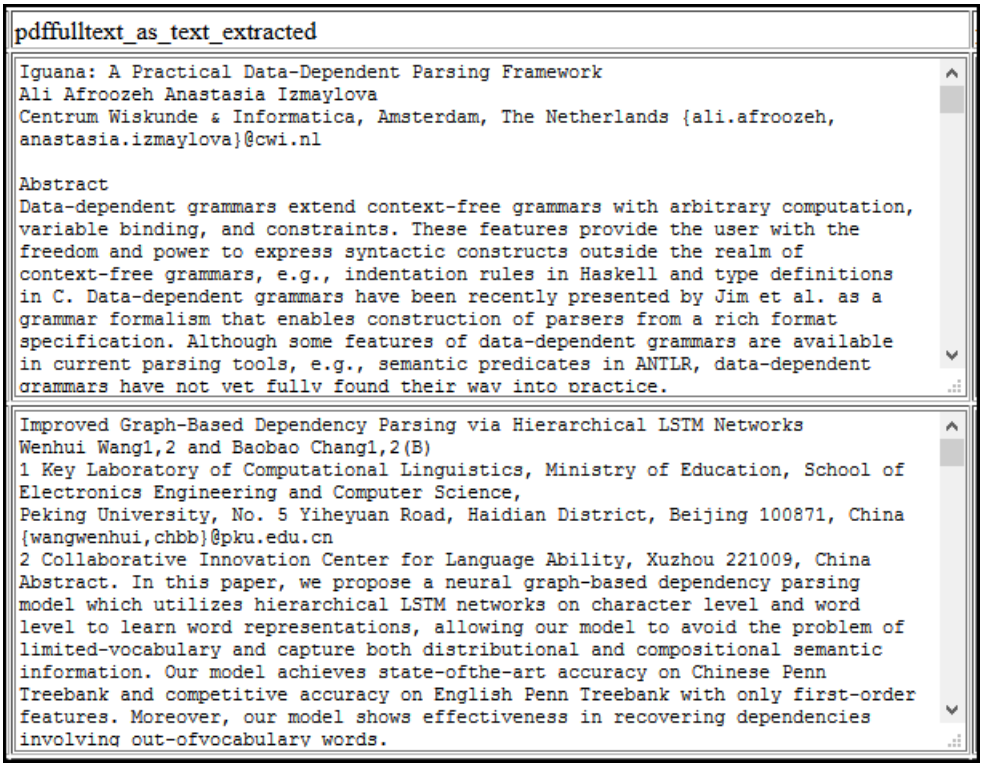


Figure 4.44 An excerpt of the input for preprocessing in the snapshot

one incident of a word that cannot be related to a natural language. Illustrated in figure 4.47 is an abstracttext that contains the word "incorporatedninto".

We choose "incorporatedninto" and correct this word by writing "incorporated into" instead as illustrated in figure 4.48.

We scroll until the end of the page and submit the modifications for abstracttexts with the button shown in figure 4.49.

Optimization for abstracttexts can be repeated as much as wanted for further optimizations after submit.

4.10 Optimize fulltexts for "Parsing"

As it is presented in section 4.9 abstracttexts should be optimized if it is assumed that abstracttexts contain words that are not part of a selected natural

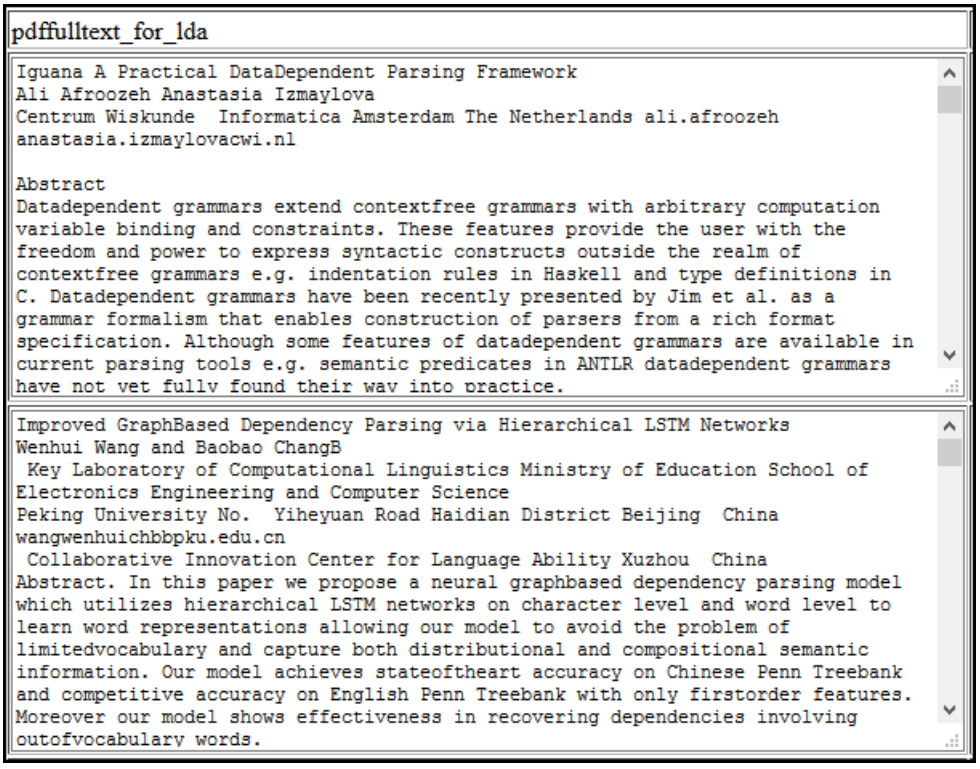


Figure 4.45 An excerpt of the preprocessing's output in the snapshot

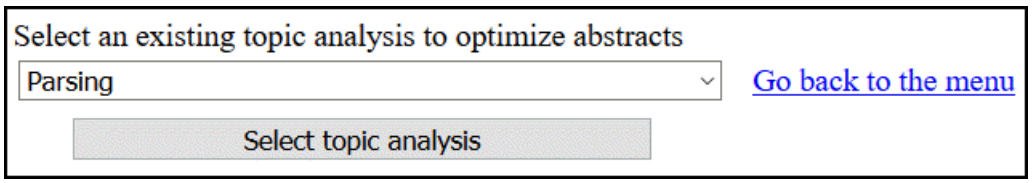


Figure 4.46 User input of "Parsing" in "topic_analysis_08_optimize_abstracts.php"

language. Fulltexts can contain words that are not part of a selected natural language. Above that, fulltexts can also contain phrases that are not a contribution for studying and summarizing "Parsing" as demanded in

- section 2.1 in terms of the systematic mapping study and
- section 2.3 in terms of topic analysis.

In order to show an example concerning this issue we select "Parsing" as illustrated in figure 4.50 in a first step in order to submit to the target page.

The target page contains an example fulltext which is writeable for optimizing as illustrated in figure 4.51.

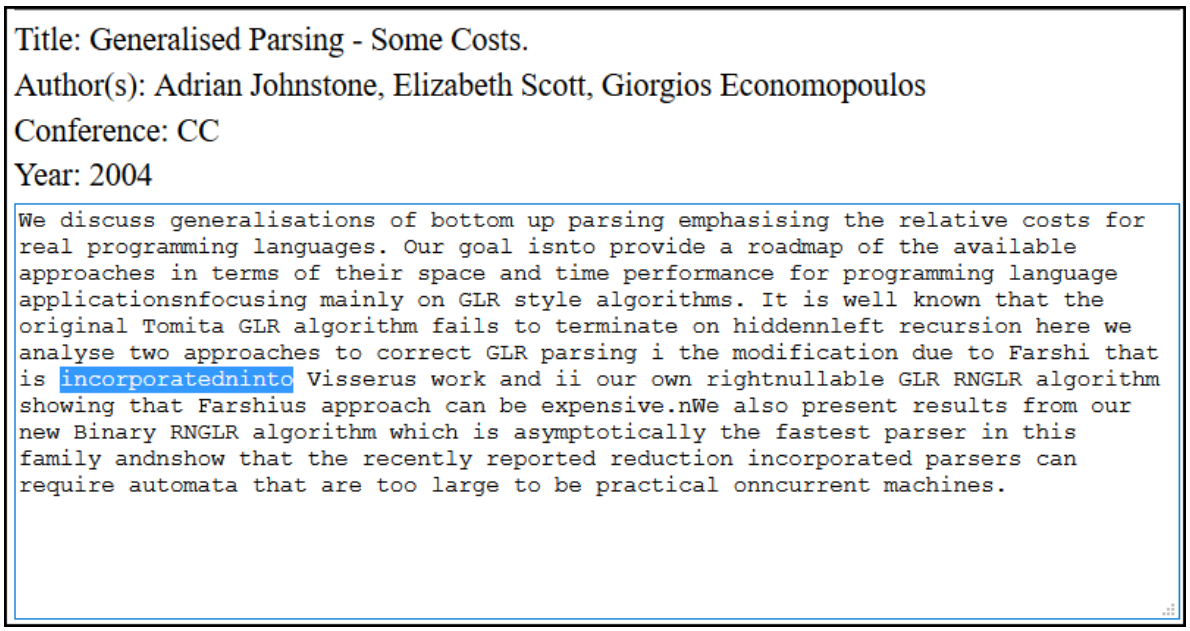


Figure 4.47 A loaded abstracttext with "incorporatedninto" as a word that is not related to a natural language

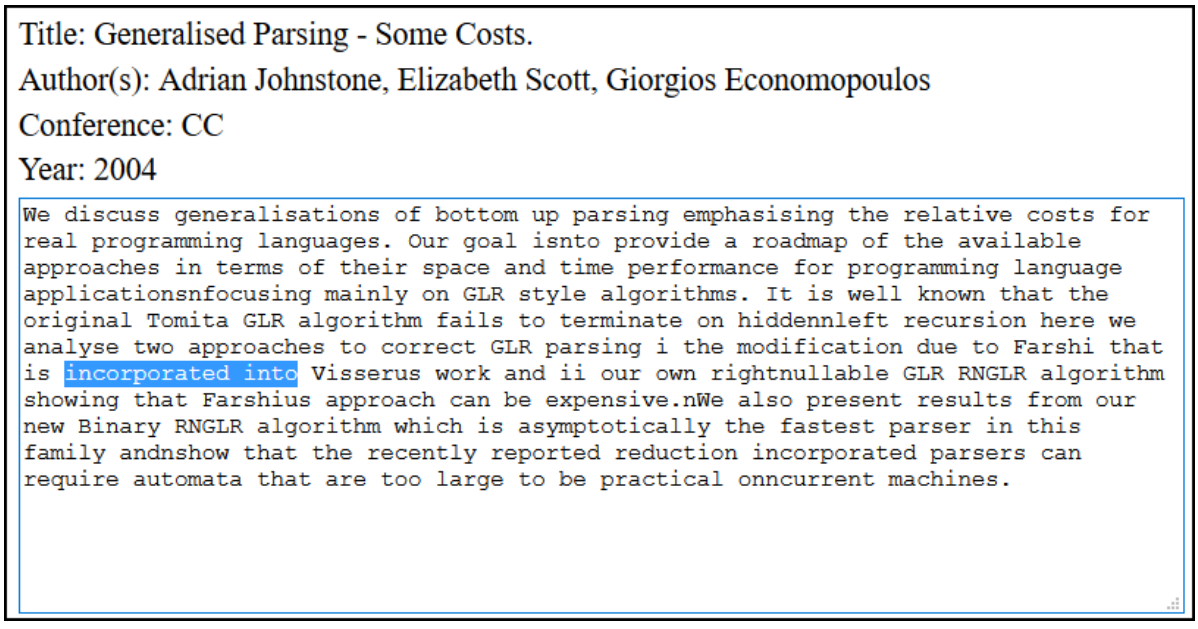


Figure 4.48 The abstracttext with the corrected word for "incorporatedninto"

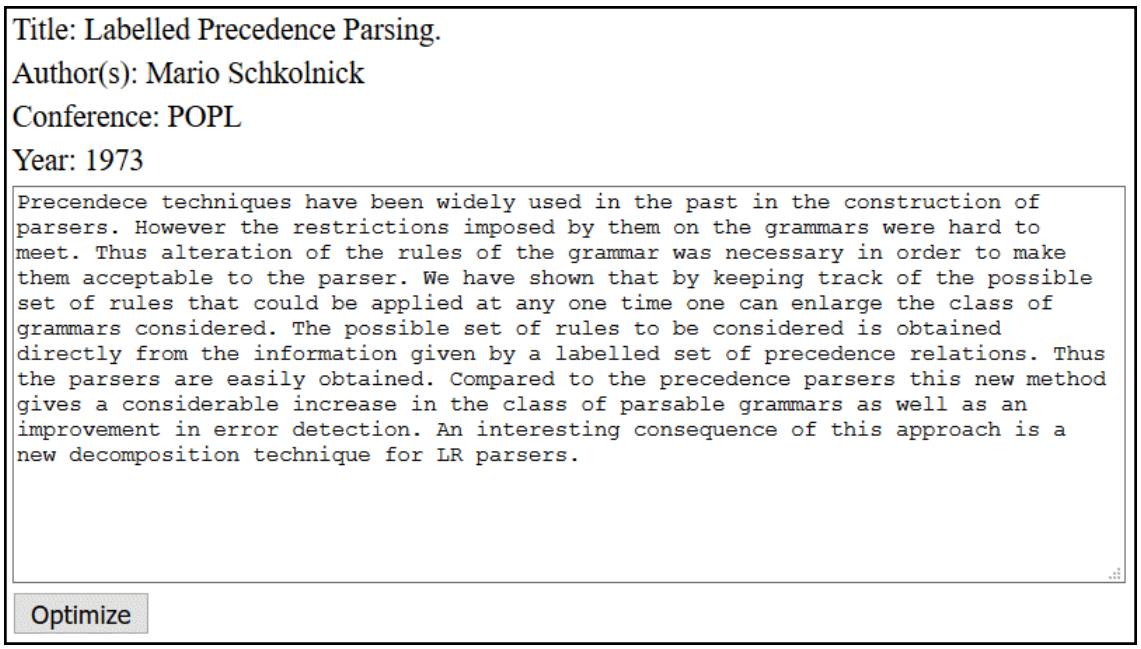


Figure 4.49 Button for submitting modifications on abstracttexts

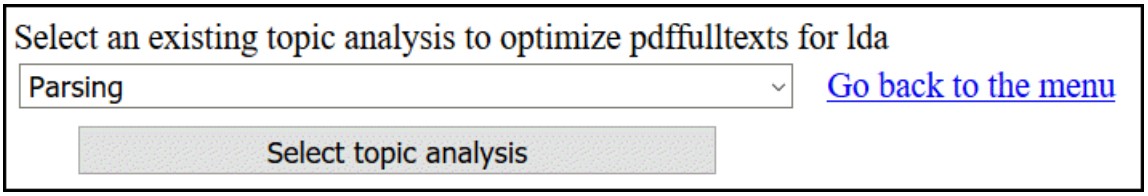


Figure 4.50 User input of "Parsing" in "topic_analysis_09_optimize_pdffulltexts.php"

This excerpt of a fulltext in figure 4.51 contains multiple words marked with the blue color that indeed are part of a natural language but do not contribute the study and summary of "Parsing". Therefore, in a second step we delete these words as shown in figure 4.52 and continue with optimizing until any fulltext just contains words that are words of a natural language and, moreover, are a contribution for studying and summarizing "Parsing". Then we activate the button with the value "Optimize" in a third step and submit the changes to the topic analysis tool.

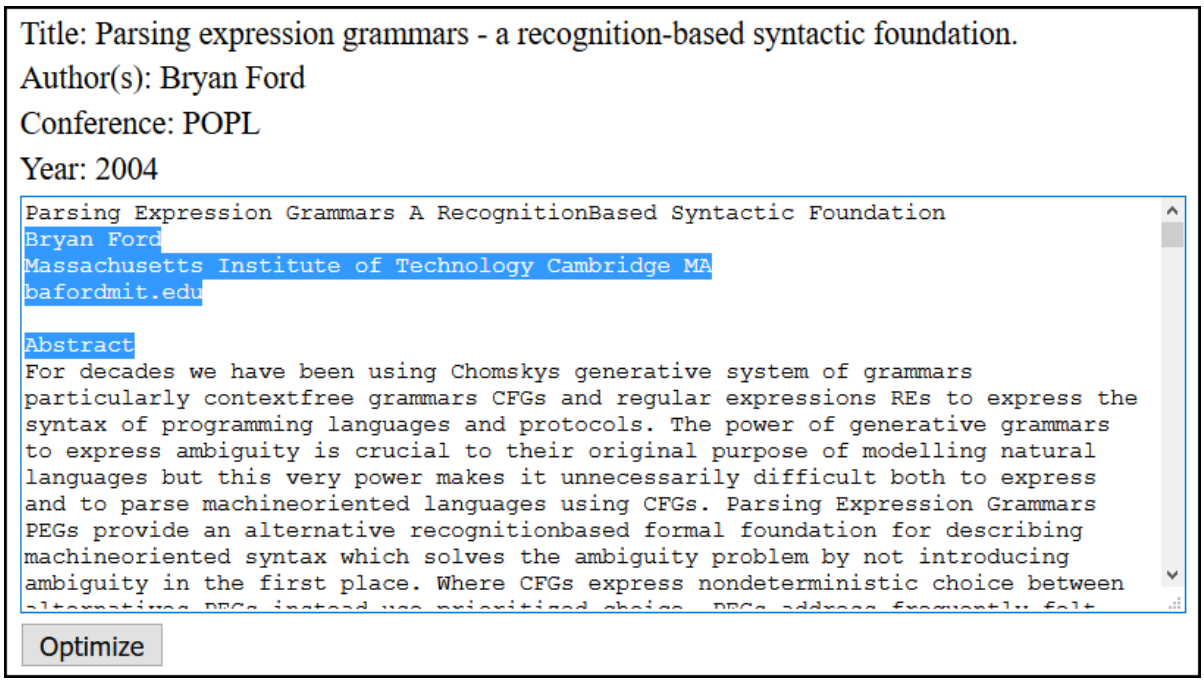


Figure 4.51 A set of words that is not a contribution for studying and summarizing "Parsing"

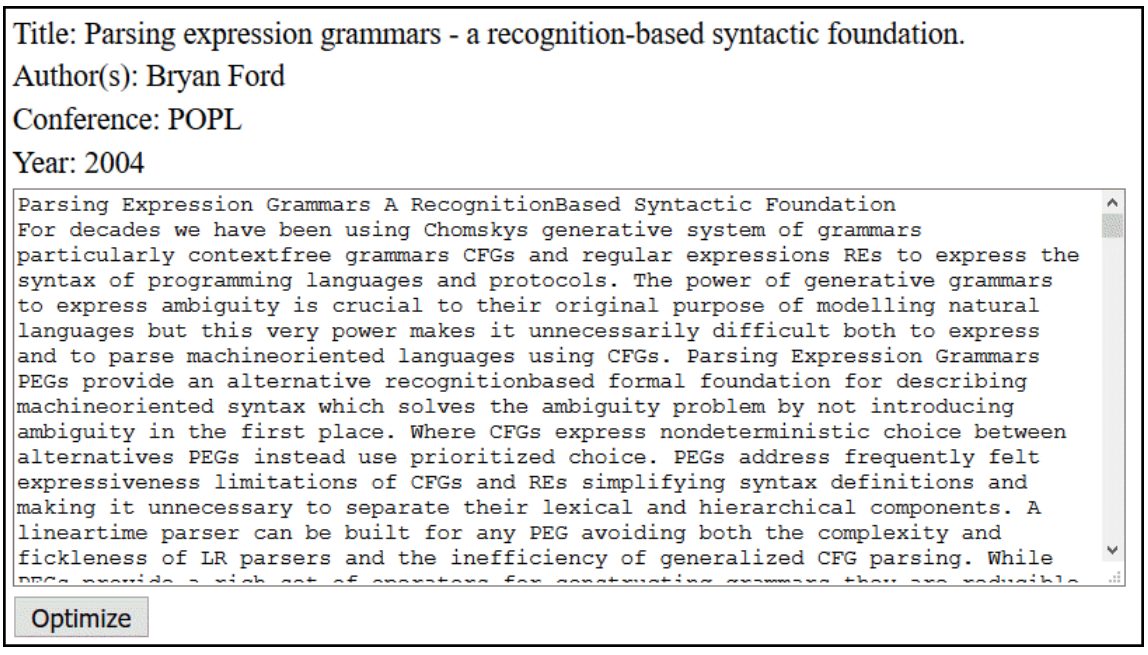


Figure 4.52 An excerpt of a fulltext that is optimized

4.11 Add and modify an environment for latent dirichlet allocation for "Parsing"

The reason for using an environment for latent dirichlet allocation is that more than one of such environments can be created for one topic analysis environment. This implies that the user can save more than one sets of input parameters for one topic analysis environment where each environment for a latent dirichlet allocation has one set of input parameters. The input parameters are for the execution of latent dirichlet allocation which is described in section 4.12.
In terms of adding a new latent dirichlet allocation environment five input fields are provided. Four input parameters are for the execution of latent dirichlet allocation and one parameter provides the name for the latent dirichlet allocation environment. The four first mentioned parameters are needed for the following two purposes:

1. Select fulltexts or abstracttexts of this topic analysis environment in terms of their membership with relation to "conference" and "year of publication" provided that these abstracttexts and fulltext are preprocessed (see section 4.7 and section 4.8) and not excluded (see section 4.6).

2. The amount of topics that the execution of latent dirichlet allocation must create.

For neither of the four input parameters for the execution of latent dirichlet allocation it is prescribed not to select no "conference" or "year of publication". This enables greatest flexibility because a selection must also consider fulltexts or abstracttexts that, as an example, are members of any "conference" and/or any "year of publication".
In order to add a new latent dirichlet allocation environment which has the name "CC" we load the script for adding a new latent dirichlet allocation environment. In this script which is shown in figure 4.53 we select the topic analysis environment for "Parsing" because we want to add latent dirichlet dirichlet allocation for this category.

After this script is submitted the empty input mask for the five input parameters is shown as illustrated in figure 4.54.

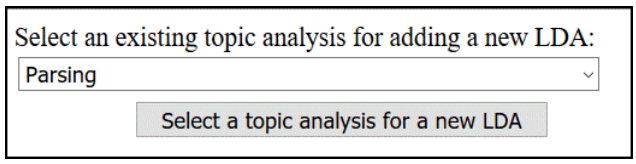


Figure 4.53 User input of "Parsing" in "topic_analysis_10_add_LDA.php"

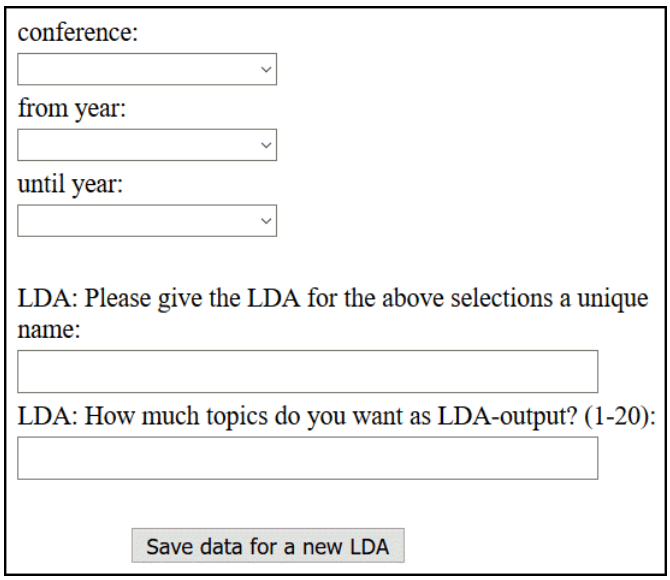


Figure 4.54 Input fields in the user mask for adding a new latent dirichlet allocation environment for "Parsing"

The input field for "conference" provides a grouped selection based on the conferences of the collected scientific articles in the topic analysis tool as illustrated in figure 4.55.

The same applies for the possible input parameters for "year from" illustrated in figure 4.56 and "year to" illustrated in figure 4.57 which is a grouped selection on any "year of publication" of scientific articles in the topic analysis tool.

With the help of the above introduced input fields we add a new latent dirichlet allocation environment for the execution of latent dirichlet allocation as follows:

- As a name for the latent dirichlet allocation environment we choose "CC".
- The amount of topics for the execution of latent dirichlet allocation for this environment is "3".

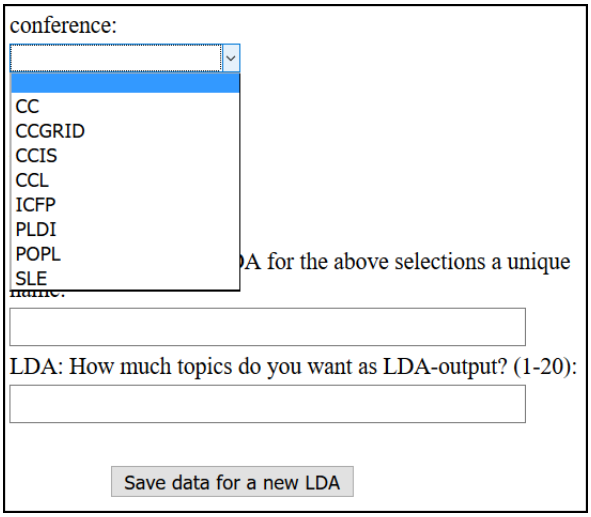


Figure 4.55 Grouped conferences based on collected scientific articles in the topic analysis tool

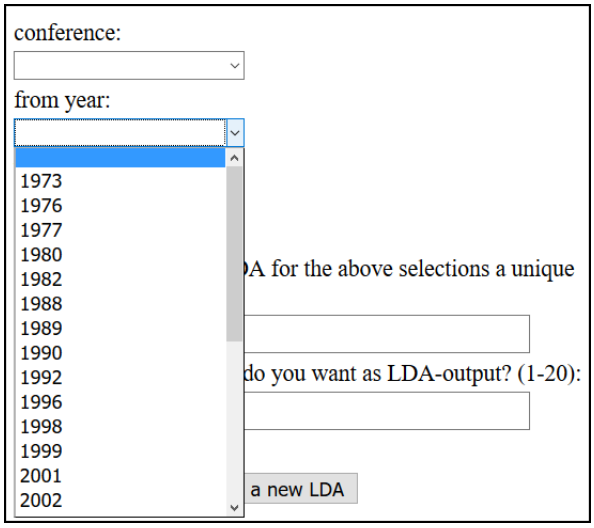


Figure 4.56 Grouped "year of publication" of any scientific article as input parameter for "year from"

- The conference is "CC".
- Any fulltext or abstracttext between 1973 and 2015 is taken into consideration.

Figure 4.58 shows the input parameters for the input form for adding a new latent dirichlet allocation environment in line of the topic analysis tool.

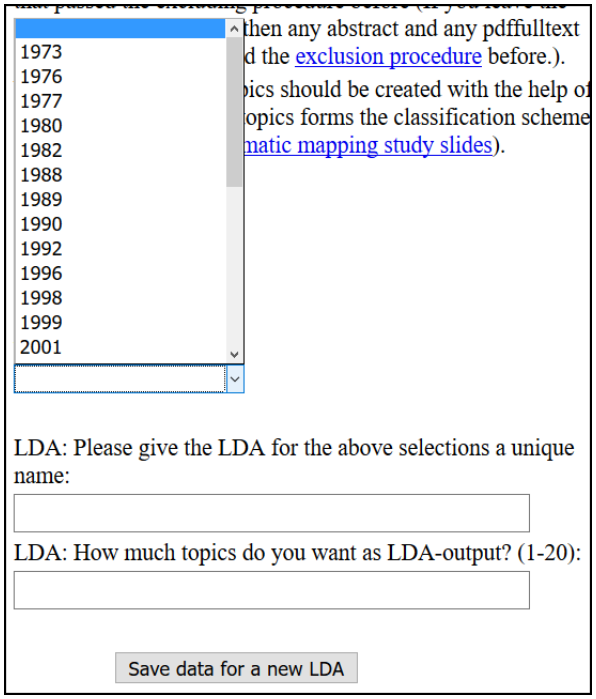


Figure 4.57 Grouped "year of publication" of any scientific article as input parameter for "year to"

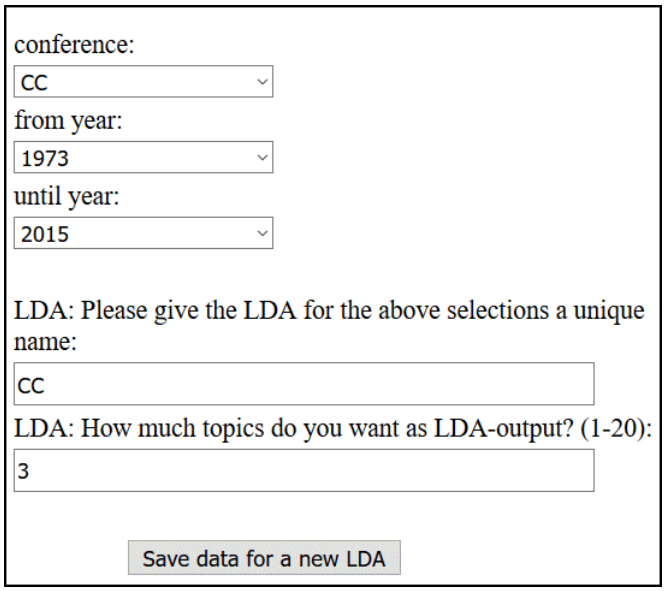


Figure 4.58 Filled input fields for input parameters for a new latent dirichlet allocation environment named "CC"

After submitting these input parameters with the help of the submit button that has the value "Save data for a new LDA" the input parameters are transmitted to the topic analysis tool.
In order to be able to modify the input parameters of the latent dirichlet allocation environment form in figure 4.58 a modification form is provided that can be accessed if the selection form for a topic analysis environment in terms of modification is loaded. This selection form is illustrated in figure 4.59.

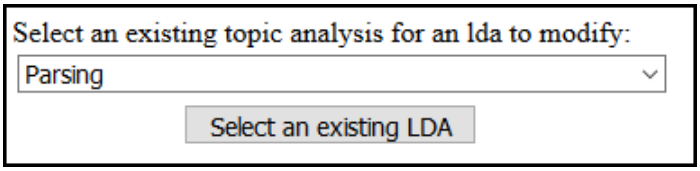


Figure 4.59 Selection of the topic analysis environment in terms of modifying "CC" in "topic_analysis_11_modify_LDA.php"

"Parsing" is selected according to figure 4.59 and submitted. Next we select the latent dirichlet allocation environment we created as illustrated in figure 4.58. This selection is shown in figure 4.60.

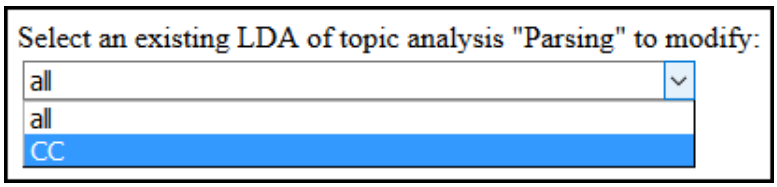


Figure 4.60 Selection of "CC" in "11_modify_LDA/topic_analysis_11_modify_LDA.php" for modifying "CC"

Then we have the same view on the latent dirichlet allocation as illustrated in figure 4.58. The view in the modification form either gives a confirmation that added inputs from before are correctly saved or the possibility to modify particular input parameter or. Also the modification of the name for the latent dirichlet allocation envrionment is possible. An activation of the submit button saves the modified values to the topic analysis tool. Figure 4.61 shows the modification form for the latent dirichlet allocation environment "CC" for "Parsing".

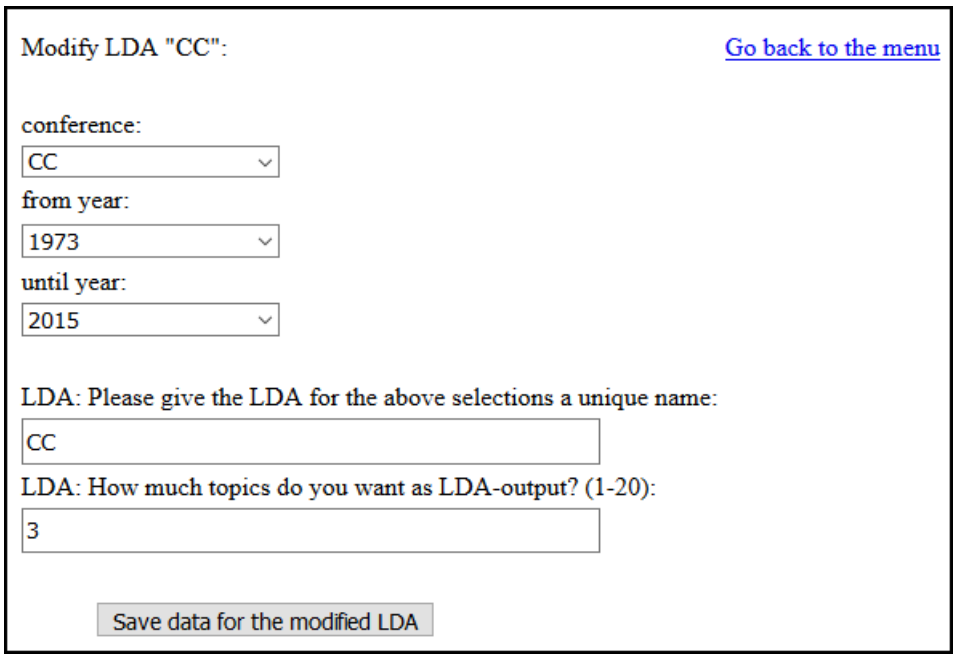


Figure 4.61 Modification form of the latent dirichlet allocation environment "CC"

4.12 Execute and delete a latent dirichlet allocation environment for "Parsing"

At first we give an introduction to the execution of an latent dirichlet allcocation environment for the topic analysis environment for "Parsing". After this introduction we show how to delete this latent dirichlet allocation environment.
In order to execute the latent dirichlet allocation environment that is added in line of the description for adding and modifying a latent dirichlet allocation environment we go to the selection page for topic analysis environments that is available for the execution of a latent dirichlet allocation environment. This selection page is illustrated in figure 4.62.

After submitting "Parsing" the latent dirichlet allocation environment "CC" that belongs to "Parsing" is selected as illustrated in figure 4.63.

The "preparation procedure for executing latent dirichlet allocation for 'CC'"[4] starts after having submitted the choice of "CC" illustrated in figure 4.63.
The "preparation procedure" for "CC" manages the following tasks:

[4]In the following we say "preparation procedure" for "preparation procedure for executing latent dirichlet allocation".

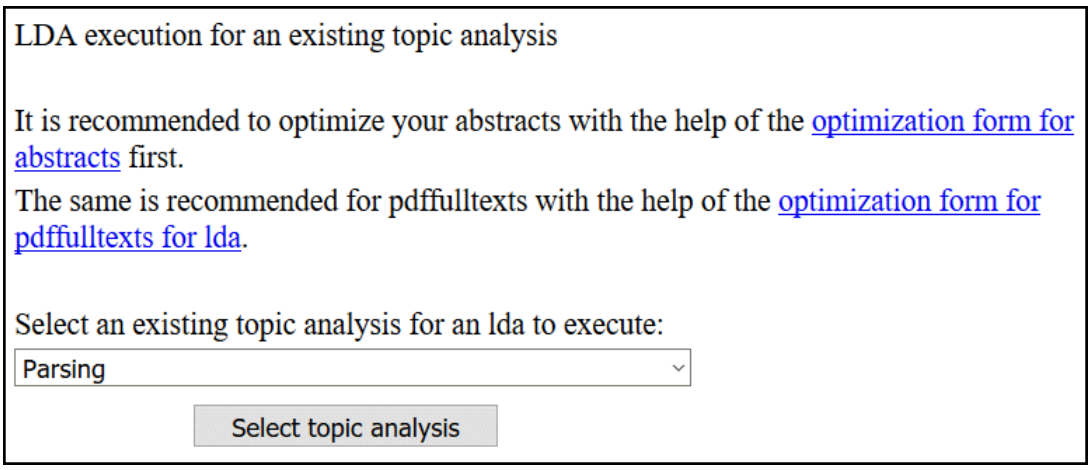


Figure 4.62 User input of "Parsing" in "topic_analysis_13_execute_LDA.php"

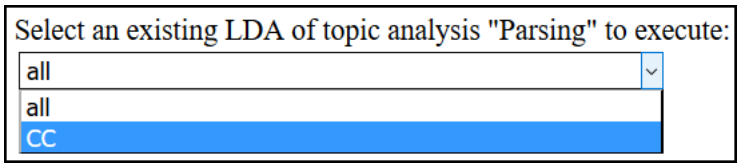


Figure 4.63 Selection of the latent dirichlet allocation environment "CC" for "Parsing" in "13_execute_LDA/topic_analysis_13_execute_LDA.php"

1. Delete the input and output files of a former latent dirichlet allocation execution for "CC".
2. SELECT the fulltexts and unless fulltexts exist SELECT the abstracttexts that are in the set of the selections for "CC" that are saved in line of the adding or modification form for "CC" (see section 4.11).
3. Write each selection from the previous step concerning a fulltext or an abstracttext to a file in the input folder.
4. Display to the user
   (a) what steps are done in line of the "preparation procedure" and
   (b) what must be done in order to execute latent dirichlet allocation with the help of R on the above mentioned input files.

The information from the "preparation procedure" refers to the selected fulltexts and abstracttexts that are transferred to the input folder of "CC". An excerpt of the

information displayed by the "preparation procedure" concerning input/output folder issues is illustrated in figure 4.64.

Execute lda "CC":

Go back to the menu

Preparing LDA execution: Copying the corpus consisting of pdffulltexts and abstracts of the selection of LDA "CC" to the inputfolder of LDA "CC" for the following attributes:

Name of the topic analysis: Parsing.

The considered conference is "CC".

The considered epoch for the publishing date for each scientific paper is from "1973" until "2015".

DELETE old files in inputfolder "C:/xampp/htdocs /topic_analysis/13_execute_LDA/lda_input_execute_output /Parsing/CC/input".

DELETE old files in outputfolder "C:/xampp/htdocs /topic_analysis/13_execute_LDA/lda_input_execute_output /Parsing/CC/output".

Copying the corpusfiles from the selection above from table "search_results" to "C:/xampp/htdocs/topic_analysis /13_execute_LDA/lda_input_execute_output/Parsing/CC/input".

Figure 4.64 Information about the "preparation procedure" referring to the handling of input and output files

The folder structure for

- "CC"'s R-script-file which is created while adding or modifying "CC" and
- "CC"'s input files which are created by this "preparation procedure"

is shown in figure 4.65 and in figure 4.66.

After this input and output information the "preparation procedure" shows that fulltexts or abstracttexts are copied to the input folder of this latent dirichlet allocation environment. The next message provides the command to execute "C:/Users/thomas/Programme/R-3.4.1/bin/i386/Rscript.exe lda.R" lda.R fetches the text corpus files consisting of the set of abstracttexts and fulltexts that is based on the selections for "conference" and "period of time based on year of publication" for "CC". The original output message how to start the execute-script based on [4] for R is shown in the illustration in figure 4.67.

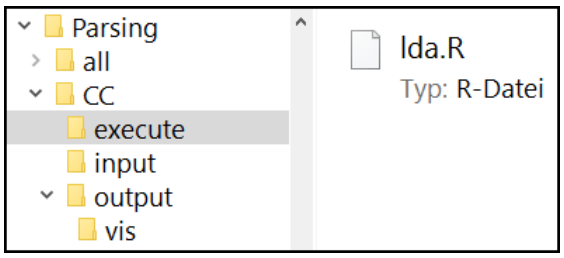


Figure 4.65 The folder structure for "CC"'s R-Script file

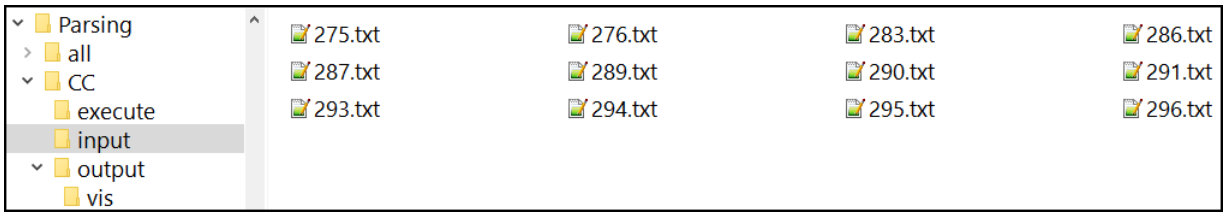


Figure 4.66 The folder structure for the input files generated by the "preparation procedure" for "CC"

Please execute lda as follows (because of timeout-issues in php if executing lda in php via C:/Users/thomas/Programme/R-3.4.1 /bin/i386/Rscript.exe):

Open a shell of your operating system (e.g. "C:/Windows /System32/cmd.exe" on Windows) in order to do the following:

Within the opened shell change to the following directory as follows: "cd C:/xampp/htdocs/topic_analysis/13_execute_LDA /lda_input_execute_output/Parsing/CC/execute".

Within the shell execute the following lda-command: "C:/Users /thomas/Programme/R-3.4.1/bin/i386/Rscript.exe lda.R".

Wait until C:/Users/thomas/Programme/R-3.4.1/bin/i386 /Rscript.exe is finished and then go to selection for displaying lda-pages in order to see and link the output-page.

Thank you for using this tool!

Figure 4.67 The "preparation procedure's" information on using the execute-script for R

The execution of "lda.R"[5] is illustrated in figure 4.68.

[5] "lda.R" is printed out in the appendix section E.1 and created with the help of R, [12], [4], [7] and [6].

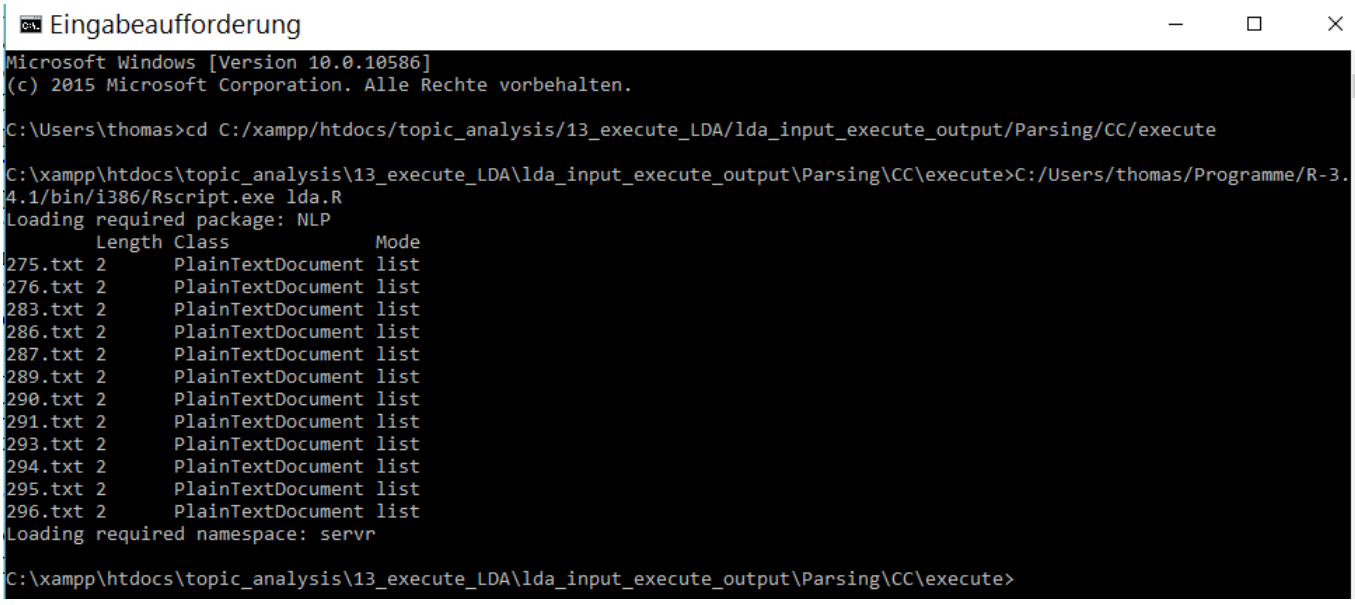


Figure 4.68 An executed R-script for "CC" with R

This execution creates output in the folder that is created while adding or modifying "CC". The folder structure for the output is illustrated in figure 4.69 and in figure 4.70.

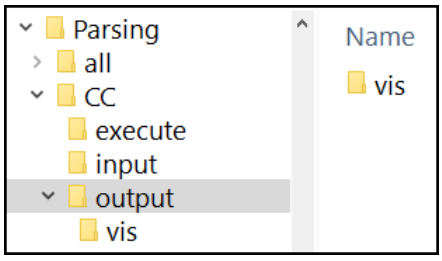


Figure 4.69 The folder structure for the output of "CC"

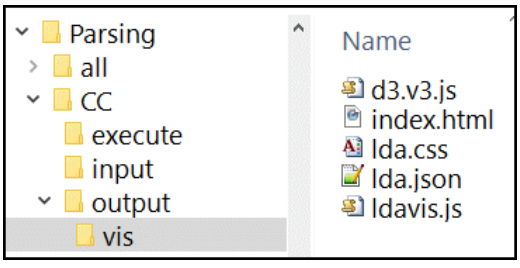


Figure 4.70 The output files of "CC"

The content of the output file in HTML is described in section 4.13.
If "CC" is not longer needed then this latent dirichlet allocation environment can be deleted with the help of the menu link "Delete existing LDA". If the topic analysis envrionment for "Parsing" and the latent dirichlet allocation environment "CC" is selected then the message in figure 4.71 is shown before "CC" with any selections will be deleted if this message is submitted.

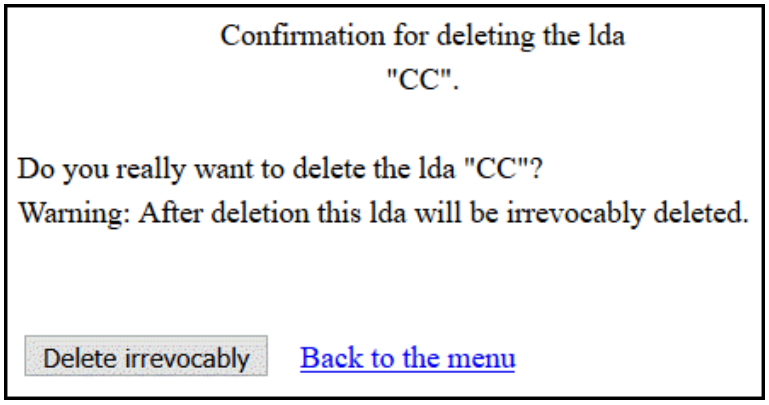


Figure 4.71 Message before submitting the deletion of "CC"

4.13 Output of an executed latent dirichlet allocation environment for "Parsing"

We create an output in line of the execution of "CC" which is a environment for latent dirichlet allocation for the topic analysis environment for "Parsing".
We state in section 4.12 that after the execution of latent dirichlet allocation in R the output files in figure 4.72 are created.

The next step is to open "index.html" in order to interpret the output. Because we work with the webbrowser we access "index.html" with the help of the topic analysis tool by activating the menu link for "displaying an existing LDA". After this link's activation we are asked to enter the topic analysis environment. Because our topic analysis environment is "Parsing" we enter "Parsing" as illustrated in in figure 4.73.

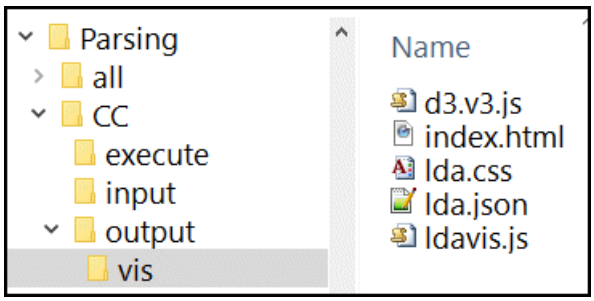


Figure 4.72 The output files for "CC"

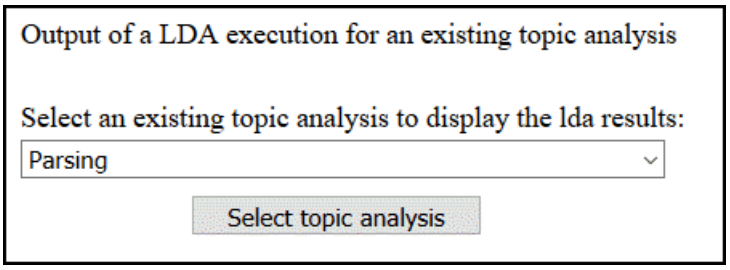


Figure 4.73 User input of "Parsing" in "topic_analysis_14_display_LDA.php"

After submitting the topic analysis environment we are asked to enter the latent dirichlet allocation environment for "Parsing". Because we describe the execution of latent dirichlet allocation for "CC" in section 4.12 we select select "CC" in the form which is illustrated in figure 4.74.

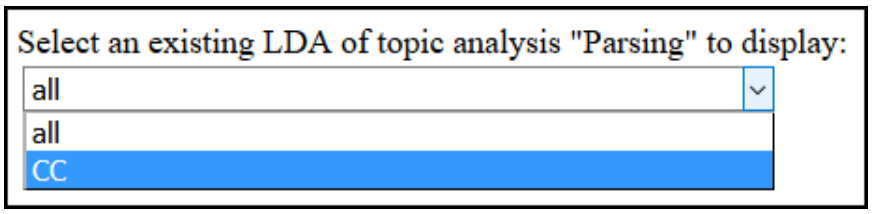


Figure 4.74 Selection of "CC" in "14_output_LDA/topic_analysis_14_output_LDA.php"

After having submitted "CC" the output page is loaded that shows the result of the latent dirichlet allocation execution for "CC" described in section 4.12. We scan this output from the top to the bottom. At the top we have the informtion illustrated in figure 4.75.

This information contains a link to the output in order to enrich information about Parsing or compiler building on other web servers with this topic analysis output.

Go back to the menu
To link this outputpage for lda "CC" please copy the link to this output to the external html-page where this output should be linked.
Output of lda "CC" of the topic analysis "Parsing" for the following selection:
The considered conference is "CC".
The considered epoch for the publishing date for each scientific paper is from "1973" until "2015".
LDA-output with the help of LDAvis is as follows:

Figure 4.75 Information about the output's parts at the top of the output page

It is sufficient to follow and copy the link in order to add it to the HTML-code on the desired web server whose content about Parsing or compiler building should be enriched. The linkable outputpage has reduced information as shown in figure 4.76.

Output of lda "CC" of the topic analysis "Parsing" for the following selection:
The considered conference is "CC".
The considered epoch for the publishing date for each scientific paper is from "1973" until "2015".
LDA-output with the help of LDAvis is as follows:

Figure 4.76 Reduced Information about the output's parts at the top of the linkable output page

The following content is the reason why we do topic analysis. We consider and analyze the studied and summarized result of latent dirichlet allocation for "CC". "CC" as an example for an environment for latent dirichlet allocation is the view on "Parsing" based on scientific articles for compiler construction conferences between 1973 and 2015. We select three topics that latent dirichlet allocation created and therefore these topics are displayed in this output as illustrated in figure 4.77.

We select and shortly analyze each topic as follows: The first topic consists of the keywords in figure 4.78.

Keywords at the top of the list for each topic have the greatest influence on the characterization of each topic because they have the highest "estimated term frequency within the selected topic" as the key is characterized at the bottom of figure 4.78 which is from [7]. The keywords with the highest "estimated term frequency within" (from [7]) in the first topic are as follows:

- parsing,

Figure 4.77 Three topics that are generated by latent dirichlet allocation for "CC"

- grammars,
- tree,
- rule,
- parser,
- nodes,
- parsers,
- reductions,
- languages,
- state,

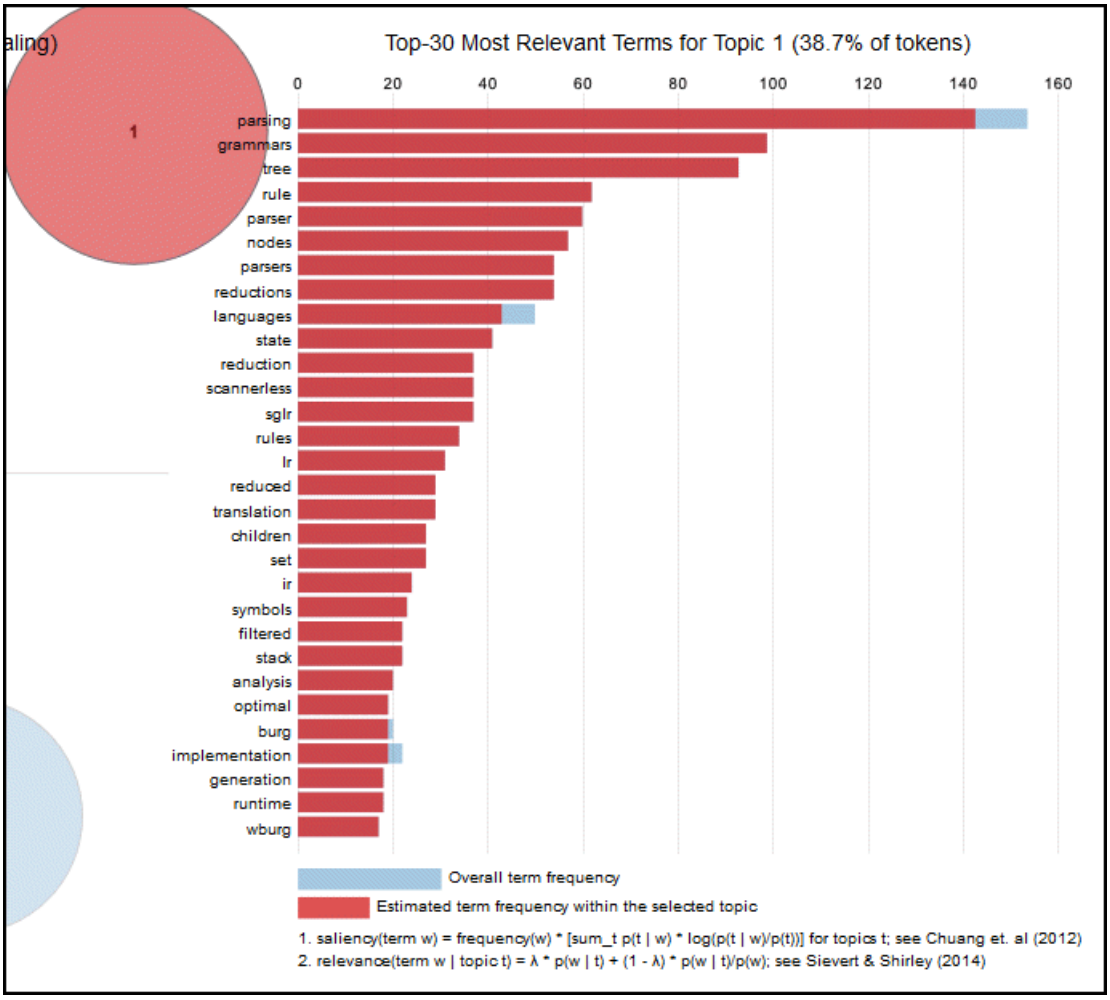


Figure 4.78 The first topic and assigned keywords to this first topic generated by latent dirichlet allocation for "CC"

- reduction,
- scannerless.

This is a topic on "Parsing" that is based on "compiler building" because it considers the fundamental parts a parser must have when supporting a compiler according to [2, page 191-312]: grammars, rule, nodes, reductions, state, etc.
Next we consider the keywords of the second topic. Because the first and the second topic have a great distance according to figure 4.77 the second topic is different to the first topic. The second topic has the following keywords according to its illustration in figure 4.79.

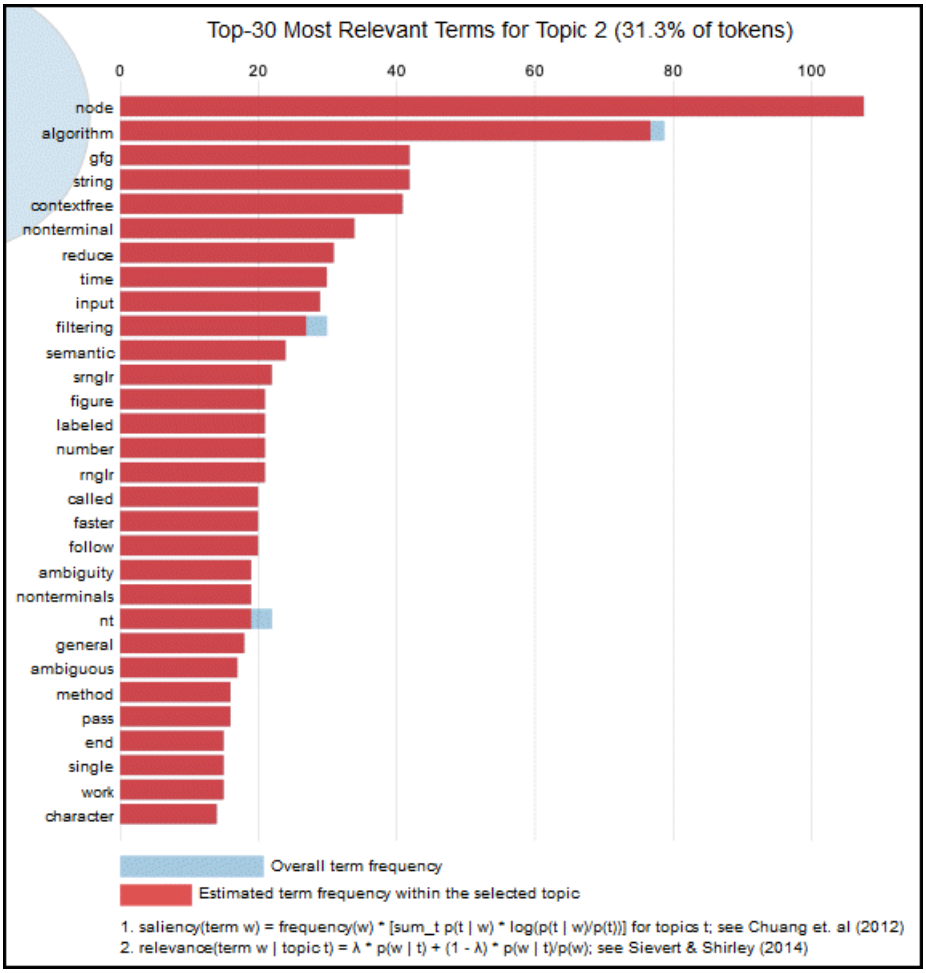


Figure 4.79 The second topic and assigned keywords to this second topic generated by latent dirichlet allocation for "CC"

Because of their most influential effect on forming the second topic according to figure 4.79 the keywords for forming the second topic are as follows:

- node,
- algorithm,
- string,
- contextfree,
- nonterminal,
- reduce,

- time,
- input,
- filtering,
- state,
- reduction,
- scannerless.

These keywords lead to the conclusion that the scientific articles for the compiler constuction conferences between 1973 and 2015 had a strong reference to programming to show that parsing is not only a topic with elements as the first topic shows but a topic that belongs or is part of a programming language.
The third topic which is illustrated in figure 4.80 contains also a topic that is helpful to understand the role of "Parsing" within a programming language. For example a tree and a table are tools for a programming language to let a program written in this programming language administrate the memory area in terms of values that are needed or provided by the processor.
As a conclusion the first topic is a general survey on Parsing in terms of [2] and topic 2 and 3 are topics with practical issues towards "Parsing".

The last section of the output presented to the user contains the scientific articles that provide the abstracttexts and fulltexts as inputs for the execution of latent dirichlet allocation. This last output's section is illustrated between figure 4.81 and figure 4.83.

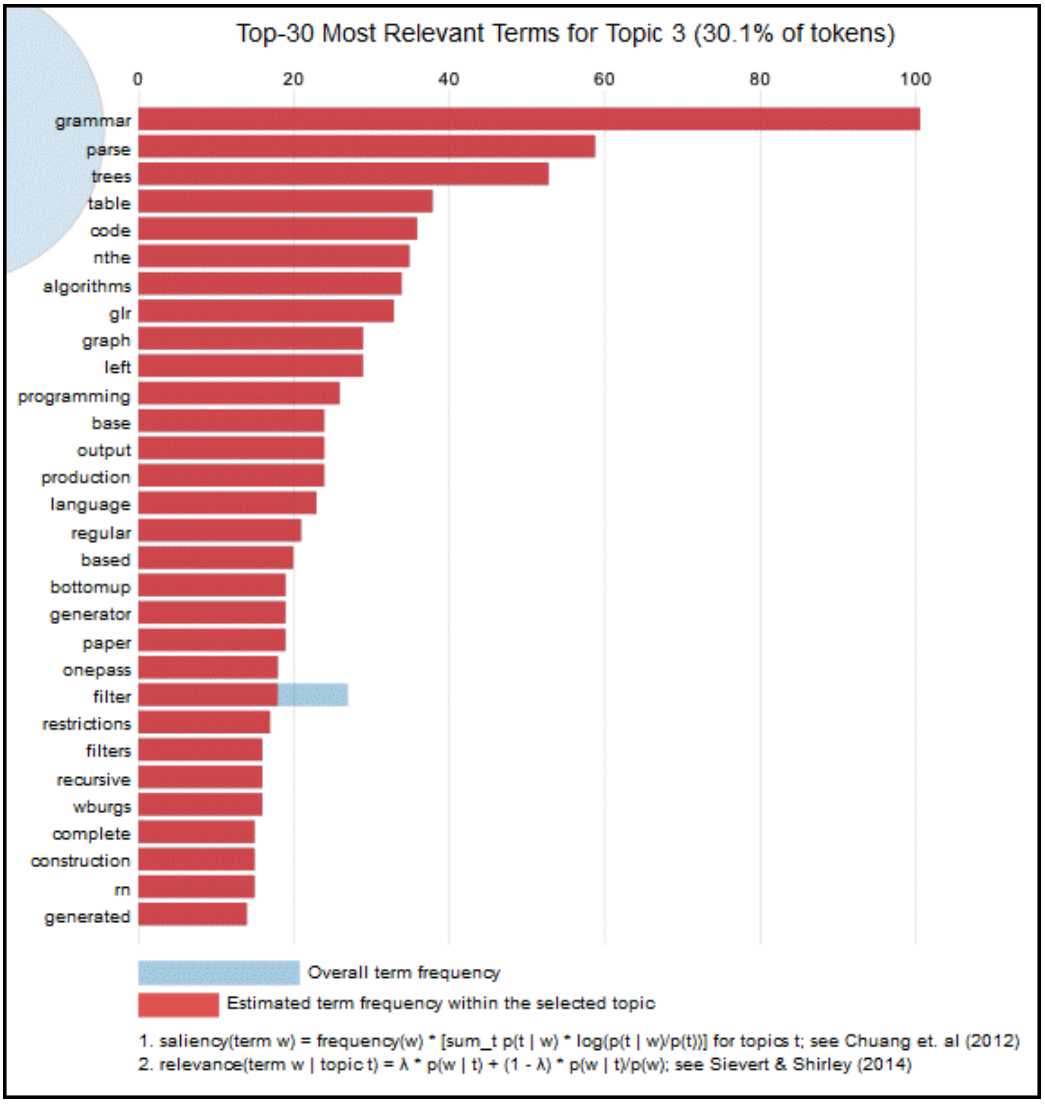


Figure 4.80 The third topic and assigned keywords to this third topic generated by latent dirichlet allocation for "CC"

The LDAvis-output from above was affected by the following scientific articles:

1) Title: A Hybrid Top-Down Parsing Technique (Abstract).
Author(s): Heinz Dobler
Conference: CC
Year: 1990

2) Title: Syntax Directed Translation with LR Parsing.
Author(s): Borivoj Melichar
Conference: CC
Year: 1992

3) Title: Attribute-Directed Top-Down Parsing.
Author(s): Karel Müller
Conference: CC
Year: 1992

4) Title: One-Pass, Optimal Tree Parsing - With Or Without Trees.
Author(s): Todd A. Proebsting, Benjamin R. Whaley
Conference: CC
Year: 1996

Figure 4.81 First part of the output's last section

5) Title: Faster Generalized LR Parsing.
Author(s): John Aycock, R. Nigel Horspool
Conference: CC
Year: 1999

6) Title: Directly-Executable Earley Parsing.
Author(s): John Aycock, R. Nigel Horspool
Conference: CC
Year: 2001

7) Title: Generalised Parsing - Some Costs.
Author(s): Adrian Johnstone, Elizabeth Scott, Giorgios Economopoulos
Conference: CC
Year: 2004

8) Title: Faster Scannerless GLR Parsing.
Author(s): Giorgios Economopoulos, Paul Klint, Jurgen J. Vinju
Conference: CC
Year: 2009

Figure 4.82 Second part of the output's last section

9) Title: Parsing C/C++ Code without Pre-processing.
Author(s): Yoann Padioleau
Conference: CC
Year: 2009

10) Title: On LR Parsing with Selective Delays.
Author(s): Eberhard Bertsch, Mark-Jan Nederhof, Sylvain Schmitz
Conference: CC
Year: 2013

11) Title: Faster, Practical GLL Parsing.
Author(s): Ali Afroozeh, Anastasia Izmaylova
Conference: CC
Year: 2015

12) Title: A Graphical Model for Context-Free Grammar Parsing.
Author(s): Keshav Pingali, Gianfranco Bilardi
Conference: CC
Year: 2015

Figure 4.83 Thirt part of the output's last section

Chapter 5

Conclusions

We summarize the work in line of this topic analysis tool described in the main and in the demo part of this thesis in section 5.1 and we suggest what could be done after finishing this thesis for this topic analysis tool in section 5.2.

5.1 Summary

In section 1 we are motivated to implement a software tool that is able to do topic analysis based on collecting, preprocessing and analyzing scientific articles.
We finish a software tool that is able to do topic analysis with the following features:

- A scientific field's category can be saved as a topic analysis environment.
- Scientific articles can be collected with the help of research questions and search strings according to systematic mapping study introduced in section 2.1 within a topic analysis environment for this scientific field's category.
- A preprocessing and an optimizing procedure lead to clean input of abstracttexts and fulltexts for the next analysis procedure according to data mining methods illustrated in section 1 in figure 1.1 which is copied from [1, page 4].
- The execution of preprocessed abstracttexts and fulltexts of scientific articles that are about a given scientific field's category is managed with the help of latent dirichlet allocation based on [6].

- The output of the analyzed abstracttexts and fulltexts of a scientific field's category based on a latent dirichlet allocation execution can be displayed on any webpage because HTML can be linked to any webpage.
- Running any functionality listed above for the scientific field's category of "Parsing" as a running example is successful.

Unfortunately, the time for extending the implemented topic analysis tool is limited. Therefore, we list future tasks for this topic analysis tool in section 5.2 that are worth to implement.
We hope that this topic analysis tool can be of great support for the purpose that a given scientific field is researched with the help of its categories.

5.2 Future works

The following points must be considered for future works:

1. The handlers that are presented in section 3.3.4 and that access a particular web search engine for fetching search results must be updated from time to time. The reason for such possible updates in the future is that returned search results from particular web search engines could contain modified data compared to search results that could be handled by the above mentioned handlers. Thus if the filling or the completion procedure gives an error message a update at the above mentioned handler's code parts will be necessary. These code parts have the issue to find the right position from search results in files from particular web search engines in order to extract the data to table "search_results".
2. The next todo for future works with regard to the implemented topic analysis tool is the implementation of the following feature of the systematic map which is the fifth step in the systematic mapping study introduced in section 2.1: This feature is for the output of latent dirichlet allocation introduced in section 3.3.11 and has the following attribute: The just mentioned output must be enriched by the information what the amount of keywords is in each scientific article that appear as a stemmed version of these keywords in let's say one of the five most influencing keywords that created a topic with the help of latent dirichlet allocation.

3. The completing-procedure of the functionality "fill_and_complete" must be an ordered map from an input-handler to an output-handler as follows:

```
input:
    array(
        title   => ...,
        authors => ...,
        year => ...,
        ...
    )
output:
    array(
        abstract => ...,
    )
```

4. The last todo according to the 14. requirement in section 3.1 is that the output described in section 3.3.11 must show a pointer to chapters or sections of fulltexts that are presented as fulltexts.

Appendix A

Glossary

Application Programming Interface "An application programming interface (API) is a set of subroutine definitions, protocols, and tools for building application software. In general terms, it is a set of clearly defined methods of communication between various software components. " (This description is quoted from [13]). 7, 45

Attributes of scientific articles The attributes of a scientific article are the name of the author(s), the title of the scientific article, the year of appearance of the scientific article, the conference that was the cause for the scientific article, the abstract and the fulltext that are completely or partly part of the search results from the web search engines and completely part of the table "search_results" that is part of the background database of this topic analysis tool. 9, 11, 25, 29, 33, 45, 103, 104

Category A category is a topic of a scientific field which beside other scientific field's categories describes this scientific field with the goal to help to define the scope of this scientific field. An example for a category is Parsing. Another category which beside "Parsing" helps to define the scope of the scientific field compiler building is code generation (compiler). 1, 5, 7, 10, 20, 35, 48, 49, 77, 104

Entity-relationship model An entity-relationship model consists of at least two entities and a relation between these two entities. The relation in the standard case is either "1:n" or "m:n" where 1 means that an attribute in the

left table that is related to 1 is unique. n means that the unique attribute in the left table is not unique in the right table. If "m:n" is considered then the same attribute in the left and the right table is not unique (For further information on entity-relationship model visit this introduction to database modeling). 15

Environment for latent dirichlet allocation An environment for latent dirichlet allocation is one of possibly many environments for latent dirichlet allocation inside the topic analysis tool related to a topic analysis environment that provides the input parameters for the computations of latent dirichlet allocation. 9, 13, 16, 20, 37, 39, 42, 54, 77, 87, 89, 103

Fill and complete Fill and complete is the filling-procedure and the completing-procedure from the topic analysis tool's functionality "04_fill_and_complete" for collecting search results at the topic analysis tool. In a first step the handler of a filling-procedure writes values to as much as possible attributes of scientific articles in table "search_results" which are e.g. "author(s)", "title", "conference", "year of publication", "abstracttext" and "fulltext". The second step will optionally take place if the filling-procedure leaves the fields of at least one scientific articles' attribute empty. In this case these empty fields are filled with handlers of the completing-procedure where each handler is related to a particular attribute. 9, 15, 26, 45, 61, 65, 103, 104

Handler A handler is a procedure that transforms scientific articles' values from a web search engine usually saved in a html-file to scientific articles' values in table "search_results" which is part of the background database of this topic analysis tool. 28, 102

Handler identifier A handler identifier is a name for a handler which is entered in the third text area field of the form for adding a new topic analysis environment or modifying an existing topic analysis environment.. 26

HTTP-GET-Variable "An associative array of variables passed to the current script via the URL parameters." (This description is quoted from [10]). 44

HTTP-POST-Variable "An associative array of variables passed to the current script via the HTTP POST method when using application/x-www-form-

urlencoded or multipart/form-data as the HTTP Content-Type in the request." (This description is quoted from [11]). 44

Latent dirichlet allocation Latent dirichlet allocation in line of this thesis is latent dirichlet allocation with the help of R, [12], [4], [7] and [6]. The execution of latent dirichlet allocation takes place within a R-Script shown in the appendix section E.1 that reads a text corpus based on preprocessed abstracttexts and fulltexts into a variable whose values are the input for natural language processing within this R-Script before it provides its output for the latent dirichlet allocation based on [6].
Adjustments for the execution of latent dirichlet allocation are a matter of the environment for latent dirichlet allocation. 2, 3, 8, 10, 13, 16, 20, 29, 32, 34, 37, 39, 44, 47, 48, 61, 66, 69, 77, 82, 87, 99, 102, 104

Paywalls "Paywalls are restricting access to Internet content via a paid subscription" (This description is quoted from [15]). 12, 56, 62

Scientific field A scientific field is an unambiguous collection of categories which describe this scientific field. An example for a scientific field is compiler building which is a branch of programming language theory where programming language theory is a branch of computer science). 1, 5, 7, 10, 48, 99, 101, 104

search result In a general sense a search result is a returned HTML-information from a web search engine. In the sense of this topic analysis tool a search result from a web search engine must be saved as a value in terms of scientific articles' values in a table called "search_results" which has scientific articles' attributes. 7, 8, 10, 20, 38, 40, 48, 55, 103, 104

search string In the general sense a search string is a string that has to be formulated in a way that it can be understood by a web search engine. If it is understood by the web search engine it will be possible that the user who submitted the search string to the web search engine gets desired search results.
In the sense of this topic analysis tool search strings are used in the above mentioned general sense if the filling- and optionally the completion procedure that are part of fill and complete are executed. Search strings for the

filling procedure are input for the second text area field when a new topic analysis environment is added or an existing topic analysis environment is modified. Search strings for the completing procedure are input for the third text area field when a new topic analysisenvironment is added or an existing topic analysis environment is modified. 2, 6–8, 10, 38, 98, 104

Semiautomation Semiautomation consists of parts that are manually and automatically controlled. 1

Snapshot A snapshot will be the output of table "search_results" to the user of the topic analysis tool if a fulfilling or completing procedure in line of fill and complete is finished. 30, 55, 63, 65, 67, 70

Topic analysis Topic analysis is the automatic study and summary of a category that belong to a scientific field. The automatic study and summary is possible because the sources for study and summary are scientific articles which are automatically retrieved from scientific web search engines, automatically preprocessed for analyzing and analyzed by latent dirichlet allocation in order to create a summary of this category that must provide a contribution for the scientific field this category belongs to. This contribution must happen in order to get a better explanation for this category's scientific field. That means the reason for topic analysis is to get a better explanation of a scientific field through topic analysis of each scientific field's category as described above before compiling each category's topic analysis result to make each result equal to the scientific field's explanation. 2

Topic analysis environment A topic analysis environment is an environment inside the topic analysis tool that contains a category of a scientific field. It also contains research questions, search strings and search results for this category. 8, 10, 15, 20, 30, 35, 37, 38, 42, 48, 49, 55, 67, 77, 87, 98, 102, 104

Values for scientific articles The values of a scientific article are the field' contents in a web search engine's search result. Values are also part of the table "search_results" that is part of the background database of this topic analysis tool. The search results from a web search engine as well as the table "search_results" have scientific articles' attributes where the above

mentioned fields are related to. Therefore, values are related to scientific articles' attributes. 9, 11, 28, 29, 102, 103

Appendix B

Introduction

B.1 Latent dirichlet allocation according to Microsoft

Latent Dirichlet Allocation

Updated: July 19, 2017

Use the Vowpal Wabbit library to perform VW LDA

Category: Text Analytics

Module Overview

You can use the **Latent Dirichlet Allocation** module to group otherwise unclassified text into a number of categories. Latent Dirichlet Allocation (LDA) is often used in natural language processing (NLP) to find texts that are similar. Another common term is *topic modeling*.

Generally speaking, LDA is not a method for classification per se, but uses a generative approach. What this means is that you don't need to provide known class labels and then infer the patterns. Instead, the algorithm generates a probabilistic model that is used to identify groups of topics. You can use the probabilistic model to classify either existing training cases, or new cases that you provide to the model as input.

A generative model can be preferable because it avoids making any strong assumptions about the relationship between the text and categories, and uses only the distribution of words to mathematically model topics.

- The theory is discussed in this paper, available as a PDF download: Online Learning for Latent Dirichlet Allocatio: Hoffman, Blei, and Bach
- The implementation in this module is based on the Vowpal Wabbit library (version 8) for LDA. For more information, see the Technical Notes section.

How to use LDA in an experiment

To use this module, you pass in a dataset that contains a column of text, either raw or preprocessed, and indicate how many categories you want to extract from the text. You can also set options for how you want punctuation handled, how large the terms are that you are extracting, and so forth. For details on how to prepare the text and configure the module, see How to Configure Latent Dirichlet Allocation.

When you run the experiment, the LDA module uses Bayes theorem to determine what topics might be associated with individual words. Words are not exclusively associated with any topics or groups; instead, each n-gram has a learned probability of being associated with any of the discovered classes.

Results

The module creates these outputs:

- The source text, together with a score for each category
- A feature matrix, containing extracted terms and coefficients for each category
- A transformation, which you can save and reapply to new text used as input

Because this module uses the Vowpal Wabbit library, it is very fast. For more information about Vowpal Wabbit, see the GitHub repository which includes tutorials and an explanation of the algorithm..

Related Tasks

See the Cortana Analytics Gallery for examples of experiments that use natural language processing in Python, feature hashing, and other text processing techniques.

How to Configure Latent Dirichlet Allocation

1. Add the **Latent Dirichlet Allocation** module to your experiment.
2. As input for the module, provide a dataset containing one or more text columns.
3. For **Target columns**, choose one or more columns containing text to analyze.

   You can choose multiple columns but they must be of the string data type.

   In general, because LDA creates a large feature matrix from the text, you'll typically analyze a single text column.
4. For **Number of topics to model**, type an integer between 1 and 1000 that indicates how many categories or topics you want to derive from the input text.

   By default, 5 topics are created.
5. For **N-grams**, specify the maximum length of N-grams generated during hashing.

   The default is 2, meaning that both bigrams and unigrams are generated.
6. Select the **Normalize** option to converting output values to probabilities. Therefore, rather than representing the transformed values as integers, values in the output and feature dataset would be transformed as follows:
   - Values in the dataset will be represented as a probability where `P(topic|document)`.
   - Values in the feature topic matrix will be represented as a probability where `P(word|topic)`.
7. Select the option, **Show all options**, and then set it to TRUE if you want to view and then set additional advanced parameters.

   These parameters are specific to the Vowpal Wabbit implementation of LDA. There are some good tutorials about LDA in Vowpal Wabbit online, as well as the official Vowpal Wabbit Wiki.

   See this sample for examples in version 8 and use of VW in Azure ML.

   - **Rho parameter**. Provide a prior probability for the sparsity of topic distributions. Corresponds to VW's `lda_rho` parameter. You would use the value 1 if you expect that the distribution of words is flat; i.e, all words are assumed equiprobable. If you think most words appear sparsely, you might set it to a much lower value.
   - **Alpha parameter**. Specify a prior probability for the sparsity of per-document topic weights. Corresponds to VW's `lda_alpha` parameter.
   - **Estimated number of documents**. Type a number that represents your best estimate of the number of documents (rows) that will be processed. This lets the module allocate a hash table of sufficient size.

     Corresponds to the `lda_D` parameter in Vowpal Wabbit
   - **Size of the batch**. Type a number that indicates how many rows to include in each batch of text sent to Vowpal Wabbit.

     Corresponds to the `batch_sz` parameter in Vowpal Wabbit.
   - **Initial value of iteration used in learning update schedule**. Specify the starting value for the learning rate.

     Corresponds to the `initial_t` parameter in Vowpal Wabbit.
   - **Power applied to the iteration during updates**. Indicate the level of power applied to the iteration count during online updates.

     Corresponds to the `power_t` parameter in Vowpal Wabbit.
   - **Number of passes over the data**. Specify the number of times the algorithm will cycle over the data.

Corresponds to the `epoch_size` parameter in Vowpal Wabbit.

8. Select the option, **Build dictionary of ngrams** or **Build dictionary of ngrams prior to LDA**, if you want to create the ngram list in an initial pass, before classifying text.

   If you create the initial dictionary beforehand, you can later use the dictionary when reviewing the model. Being able to map results to text rather than numerical indices is generally easier for interpretation. However, saving the dictionary will take longer and use additional storage.

9. For **Maximum size of ngram dictionary**, type the total number of rows that can be created in the ngram dictionary.

   This option is useful for controlling the size of the dictionary. However, if the number of ngrams in the input exceeds this size, collisions may occur.

10. Run the experiment.

Results

The module has two outputs:

- **Transformed dataset**. Contains the input text, and a specified number of discovered categories, together with the scores for each text example for each category.
- **Feature topic matrix**. The leftmost column contains the extracted text feature, and there is a column for each category containing the score for that feature in that category.

For details and an example based on customer review text, see Understanding LDA Results.

Examples

For examples of how text analytics, see these experiments in the Model Gallery:

- The Execute Python Script sample uses natural language processing in Python to clean and transform text.

Technical Notes

By default, the distributions of outputs for transformed dataset and feature-topic matrix are normalized as probabilities.

- The transformed dataset is normalized as the conditional probability of topics given a document. In this case, the sum of each row equals 1.
- The feature-topic matrix is normalized as the conditional probability of words given a topic. In this case, the sum of each column equals 1.

Occasionally the module might return an empty topic, which is most often caused by the pseudo-random initialization of the algorithm.

If this happens, you can try changing related parameters, such as the maximum size of the N-gram dictionary or the number of bits to use for feature hashing.

More About Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is often used for *content-based topic modeling*, which basically means learning categories from unclassified text. In content-based topic modeling, a topic is a distribution over words.

For example, assume that you have provided a corpus of customer reviews that includes many, many products. The text of reviews that have been submitted by many customers over time would contain many terms, some of which are used in multiple topics.

A **topic** that is identified by the LDA process might represent reviews for an individual Product A, or it might represent a group of product reviews. To LDA, the topic itself is just a probability distribution over time for a set of words.

Terms are rarely exclusive to any one product, but can refer to other products, or be general terms that apply to everything (“great”, “awful”). Other terms might be noise words. However, it is important to understand that the LDA method does not purport to capture all words in the universe, or to understand how words are related, aside from proabilities of co-occurrence. It can only group words that were used in the target domain.

After the term indexes have been computed, individual rows of text are compared using a distance-based similarity measure, to determine whether two pieces of text are like each other. For example, you might find that the product has multiple names that are strongly correlated. Or, you might find that strongly negative terms are usually associated with a particular product. You can use the similarity measure both to identify related terms and to create recommendations.

Interpreting LDA Results

To illustrate how the **Latent Dirichlet Allocation** module works, the following example applies LDA with the default settings to the Book Review dataset provided in Azure Machine Learning Studio. The dataset contains a rating column, as well as the full comment text provided by users.

Sample Source Text

This table shows only a few representative examples.

During processing, the **Latent Dirichlet Allocation** module both cleans and analyzes the text, based on parameters you specify. For example, it can automatically tokenize the text and remove punctuation, and at the same time find the text features for each topic.

text
This book has its good points. If anything, it helps you put into words what you want from a supervisor....
I admit, I haven't finished this book. A friend recommended it to me as I have been having problems with insomnia...
Poorly written I tried reading this book but found it so turgid and poorly written that I put it down in frustration. ...
Since borrowing a dog-eared copy from friends who were passing it around a number of years ago, I have not been able to get my hands on this book which became a short-lived cult favorite
The plot of this book was interesting, and it could have been a good book. Unfortunately, it wasn't. The main problem for me was that ...

Transformed Sample Data

The following table contains the transformed dataset, based on the Book Review sample. The output contains the input text, and a specified number of discovered categories, together with the scores for each category.

In this example, we used the default value of 5 for **Number of topics to model**. Therefore, the LDA module creates five categories, which we can assume will correspond roughly with the original five-scale rating system.

The module also assigns a score to each entry for each of the five categories that represent topics. A score indicates the probability that the row should be assigned to a particular category.

Movie name	**Topic 1**	**Topic 2**	**Topic 3**	**Topic 4**	**Topic 5**
this book has its good points	0.001652892	0.001652892	0.001652892	0.001652892	0.9933884

Movie name	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
friend recommended it to me	0.00198019	0.001980198	0.9920791	0.001980198	0.001980198
tried reading this book	0.002469135	0.002469135	0.9901233	0.002469135	0.002469135
borrowed it from friend	0.9901232	0.002469135	0.002469135	0.002469135	0.002469135
plot of this book was interesting	0.001652892	0.001652892	0.9933884	0.001652892	0.001652892

Feature Topic Matrix

The other output of the module is the **feature topic matrix**. This is a tabular dataset that contains the *featurized text*, , in column **Feature**, along with a score for each of the categories, in the remaining columns *Topic 1*, *Topic 2*, ...*Topic N*. The score represents the coefficient.

Feature	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
interesting	0.0240282071983144	0.0354678954779375	0.363051866576914	0.0276637824315893	0.6606635761495
was	0.0171478729532397	0.0823969031108669	0.00452966877950789	0.0408714510319233	0.0250773226897
from	0.0148224220349217	0.0505086981492109	0.00434423322461094	0.0273389126293824	0.0171484355106
plot	0.0227415889348212	0.0408709456489325	0.182791041345191	0.086937090812819	1 0.0169680136708
reading	0.0227415889348212	0.0408709456489325	0.182791041345191	0.0869370908128191	0.0169680136708
tried	0.0269724979147211	0.039026263551767	0.00443749106785087	0.0628829816088284	0.0235340728818
me	0.0262656945140134	0.0366941302751921	0.00656837975179138	0.0329214576160066	0.0214121851106
to	0.0141026103224462	0.043359976919215	0.00388640531859447	0.0305925953440055	0.0228993750526
it	0.0264490547105951	0.0356674440311847	0.00541759897864314	0.0314539386250293	0.0140606468587
friend	0.0135971322960941	0.0346118171467234	0.00434999437350706	0.0666507321888536	0.0181568637793
points	0.0227415889348212	0.0396233855719081	0.00404663601474112	0.0381156510019025	0.0337788009496
good	0.651813073836783	0.0598646397444108	0.00446809691985617	0.0358975694646062	0.0138989124411
its	0.0185385588647078	0.144253986783184	0.00408876416453866	0.0583049240441475	0.0154428055668
of	0.0171416780245647	0.0559361180418586	0.0100633904544953	0.087093930106723	0.0182573833869
borrowed	0.0171416780245647	0.0559361180418586	0.0100633904544953	0.087093930106723	0.0182573833869
has	0.0171416780245647	0.0559361180418586	0.0100633904544953	0.087093930106723	0.0182573833869
book	0.0143157047920681	0.069145948535052	0.184036340170983	0.0548757337823903	0.0156837976985
recommended	0.0161486848419689	0.0399143326399534	0.00550113530229642	0.028637149142764	0.0147675139039
this	0.0161486848419689	0.0399143326399534	0.00550113530229642	0.028637149142764	0.0147675139039

LDA Transformation

The module also outputs the *transformation* that applies LDA to the dataset, as an ITransform interface.

You can save this transformation and re-use it for other datasets. This might be useful if you have trained on a large corpus and want to reuse the coefficients or categories.

Refining an LDA Model

Because each task has unique requirements and each corpus has different characteristics in terms of the distribution of terms, typically you cannot create a single LDA model that will meet all needs.

Instead, we recommend that you try changing the model parameters, use visualization to understand the results, and get the feedback of subject matter experts to ascertain whether the topics are useful.

Measure Accuracy and Coverage

Qualitative measures can also be useful for assessing the results. Measures often used to evaluate topic modeling include:

- **Accuracy**. Are similar items really similar?
- **Diversity**. Can the model discriminate between similar items when required for the business problem?
- **Scalability**. Does it work on a wide range of text categories or only on a narrow target domain?

Refine Input Text

The accuracy of models based on LDA can often be improved by using natural language processing to clean, summarize and simplify, or categorize text. For example, the following techniques, all supported in Azure Machine Learning, might improve classification accuracy:

- Stop word removal
- Case normalization
- Lemmatization or stemming
- Named entity recognition

For more information, see Preprocess Text and Named Entity Recognition.

You might also use R or Python libraries for pre-processing of text, by using the Execute R Script or Execute Python Script modules.

Expected Inputs

Name	Type	Description
Dataset	Data Table	Input dataset

Module Parameters

Name	Type	Range	Optional	Default	Description

Name	Type	Range	Optional	Default	Description
Number of hash bits	Integer	[1;31]	Applies when the **Show all options** checkbox is *not* selected	12	Number of bits to use for feature hashing
Target column(s)	Column Selection		Required	StringFeature	Target column name or index
Number of topics to model	Integer	[1;1000]	Required	5	Model the document distribution against N topics
N-grams	Integer	[1;10]	Required	2	Order of N-grams generated during hashing
Normalize	Boolean		Required	true	Normalize output to probabilities. The transformed dataset will be P(topic|document) and the feature topic matrix will be P(word|topic).
Show all options	Boolean	True False	Required	False	Presents additional parameters specific to Vowpal Wabbit online LDA
Rho parameter	Float	[0.00001;1.0]	Applies when the **Show all options** checkbox is selected	0.01	Rho parameter
Alpha parameter	Float	[0.00001;1.0]	Applies when the **Show all options** checkbox is selected	0.01	Alpha parameter
Estimated number of documents	Integer	[1;int.MaxValue]	Applies when the **Show all options** checkbox is selected	1000	Estimated number of documents (Corresponds to lda_D parameter)
Size of the batch	Integer	[1;1024]	Applies when the **Show all options** checkbox is selected	32	Size of the batch
Initial value of iteration used in learning rate update schedule	Integer	[0;int.MaxValue]	Applies when the **Show all options** checkbox is selected	0	Initial value of iteration count used in learning rate update schedule (Corresponds to initial_t parameter)
Power applied to the iteration during updates	Float	[0.0;1.0]	Applies when the **Show all options** checkbox is selected	0.5	Power applied to the iteration count during online updates (Corresponds to power_t parameter)
Number of training iterations	Integer	[1;1024]	Applies when the **Show all options** checkbox is selected	25	Number of training iterations
Build dictionary of ngrams	Boolean	True False	Applies when the **Show all options** checkbox is *not* selected	True	Builds a dictionary of ngrams prior to computing LDA. Useful for model inspection and interpretation

Name	Type	Range	Optional	Default	Description
Number of bits to use for feature hashing	Integer	[1;31]	Applies when the option **Build dictionary of ngrams** is False	12	Number of bits to use during feature hashing
Maximum size of ngram dictionary	Integer	[1;int.MaxValue]	Applies when the option **Build dictionary of ngrams** is True	20000	Maximum size of the ngrams dictionary. If number of tokens in the input exceed this size, collisions may occur
Build dictionary of ngrams prior to LDA	Boolean	True False	Applies when the **Show all options** checkbox is selected	True	Builds a dictionary of ngrams prior to LDA. Useful for model inspection and interpretation
Maximum number of ngrams in dictionary	Integer	[1;int.MaxValue]	Applies when the option **Build dictionary of ngrams** is True and the **Show all options** checkbox is selected	20000	Maximum size of the dictionary. If number of tokens in the input exceed this size, collisions may occur

Outputs

Name	Type	Description
Transformed dataset	Data Table	Output dataset
Feature topic matrix	Data Table	Feature topic matrix produced by LDA
LDA transformation	ITransform interface	Transformation that applies LDA to the dataset

Exceptions

Exception	Description
Error 0002	Exception occurs if one or more specified columns of data set couldn't be found.
Error 0003	Exception occurs if one or more of inputs are null or empty.
Error 0004	Exception occurs if parameter is less than or equal to specific value.
Error 0017	Exception occurs if one or more specified columns have type unsupported by current module.

See Also

Text Analytics

Appendix C

Design

C.1 Information about the database of the topic analysis tool

Database „topic_analysis“

List of tables

lda

(Physical Name: lda)

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
id_topic_analysis (FK)	id_topic_analysis	INTEGER		NOT NULL
name	name	LONGVARCHAR		NOT NULL
number_of_topics_to_output	number_of_topics_to_output	INTEGER		NOT NULL
conference_selected	conference_selected	LONGVARCHAR		NOT NULL
year_from_selected	year_from_selected	LONGVARCHAR		NOT NULL
year_to_selected	year_to_selected	LONGVARCHAR		NOT NULL
dirname	dirname	LONGVARCHAR		NOT NULL

References

- topic_analysis through (id_topic_analysis)

Referenced By

- lda_id_search_results referencing (id)

lda_id_search_results

(Physical Name: lda_id_search_results)

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
id_lda (FK)	id_lda	INTEGER		NOT NULL
id_search_result (FK)	id_search_result	INTEGER		NOT NULL

References

- lda through (id_lda)
- search_results through (id_search_result)

research_questions

(Physical Name: research_questions)

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
id_topic_analysis (FK)	id_topic_analysis	INTEGER		NOT NULL
name	name	LONGVARCHAR		NOT NULL

References

- topic_analysis through (id_topic_analysis)

search_results

(Physical Name: search_results)

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
id_search_strings (FK)	id_search_strings	INTEGER		NOT NULL
exclude	exclude	BIT		NOT NULL
authors	authors	LONGVARCHAR		NOT NULL
title	title	LONGVARCHAR		NOT NULL
conference	conference	LONGVARCHAR		NOT NULL
year	year	DATE		NOT NULL
first_link_to_abstracttext	first_link_to_abstracttext	LONGVARCHAR		NOT NULL
abstracttext	abstracttext	CLOB		NOT NULL
abstracttext_for_lda	abstracttext_for_lda	CLOB		NOT NULL
first_link_to_pdffulltext	first_link_to_pdffulltext	LONGVARCHAR		NOT NULL
path_to_pdffulltext	path_to_pdffulltext	LONGVARCHAR		NOT NULL

pdffulltext_as_text	pdffulltext_as_text	CLOB		NOT NULL
pdffulltext_as_text_extracted	pdffulltext_as_text_extracted	CLOB		NOT NULL
pdffulltext_for_lda	pdffulltext_for_lda	CLOB		NOT NULL
exclusion_already_done	exclusion_already_done	BIT		NOT NULL
preprocessing_abstracttext_already_done	preprocessing_abstracttext_already_done	BIT		NOT NULL
preprocessing_pdffulltext_already_done	preprocessing_pdffulltext_already_done	BIT		NOT NULL

References

- search_strings through (id_search_strings)

Referenced By

- lda_id_search_results referencing (id)

search_strings

(Physical Name: search_strings)

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
id_topic_analysis (FK)	id_topic_analysis	INTEGER		NOT NULL
name	name	LONGVARCHAR		NOT NULL
htmlsource	htmlsource	LONGVARCHAR		NOT NULL

References

- topic_analysis through (id_topic_analysis)

Referenced By

- search_results referencing (id)

search_strings_for_results

(Physical Name: search_strings_for_results)

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
id_topic_analysis (FK)	id_topic_analysis	INTEGER		NOT NULL
name	name	LONGVARCHAR		NOT NULL

References

- topic_analysis through (id_topic_analysis)

topic_analysis

(Physical Name: topic_analysis)

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
name	name	LONGVARCHAR		NOT NULL

Referenced By

- lda referencing (id)
- search_strings referencing (id)
- search_strings_for_results referencing (id)
- research_questions referencing (id)

Appendix D

Implementation

D.1 Functionality for collecting search results

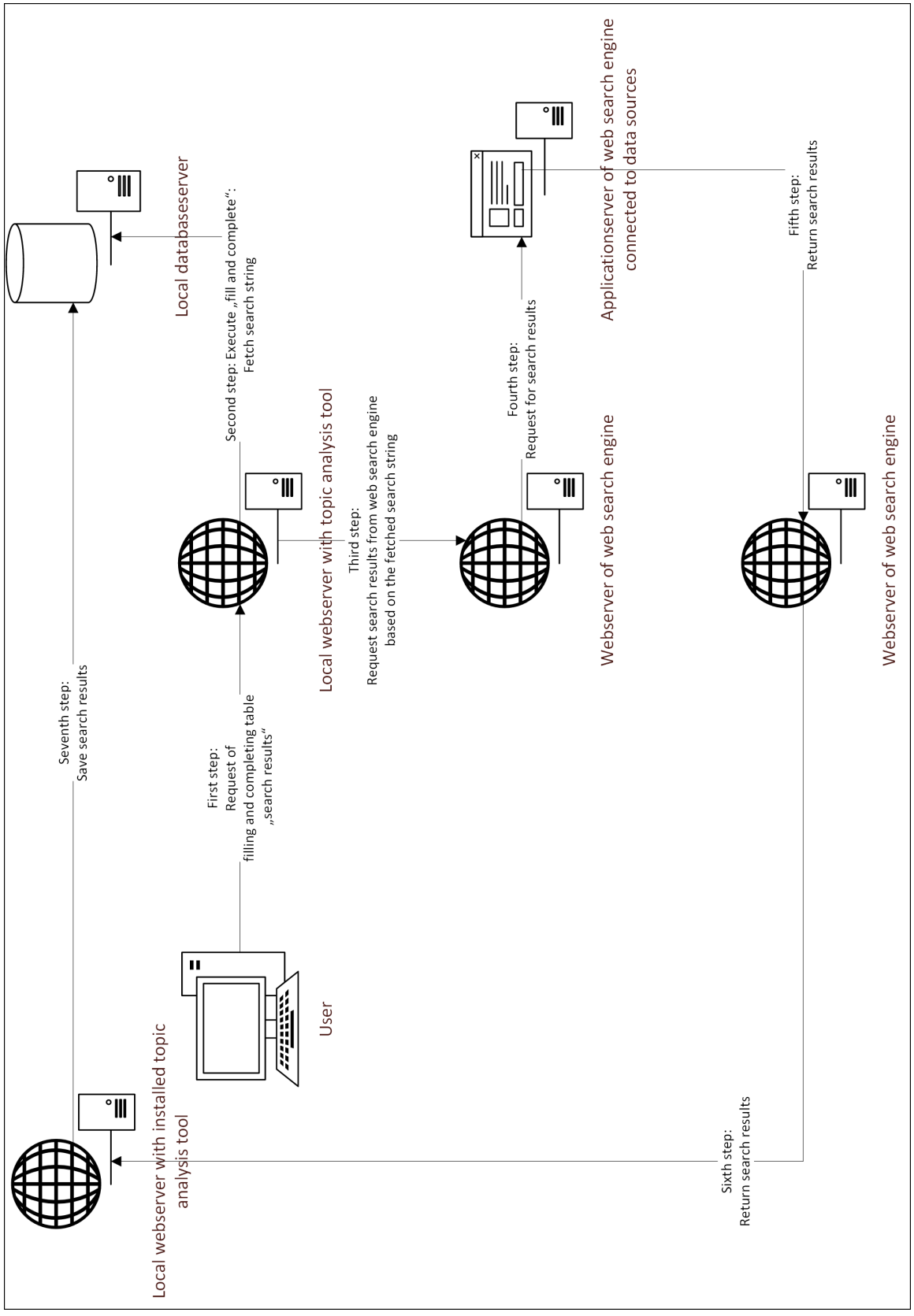


Figure D.1 Information exchange between topic analysis tool and a requested web search engine

Figure D.2 Fill and complete with the filling-procedure for `http://dblp.org/search/publ/api`

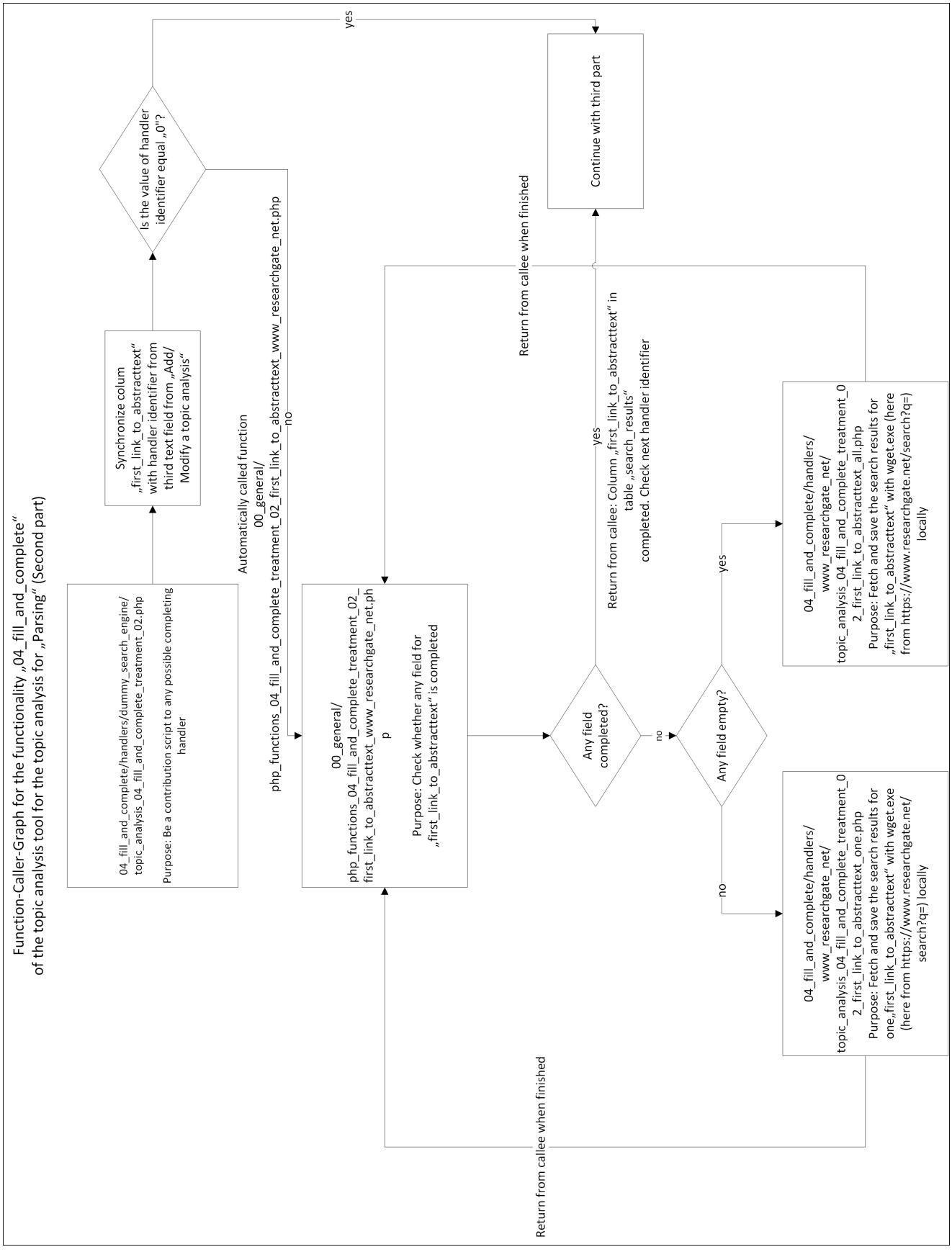


Figure D.3 Fill and complete with the completing-procedure for https://www.researchgate.net/search?q=[$title]

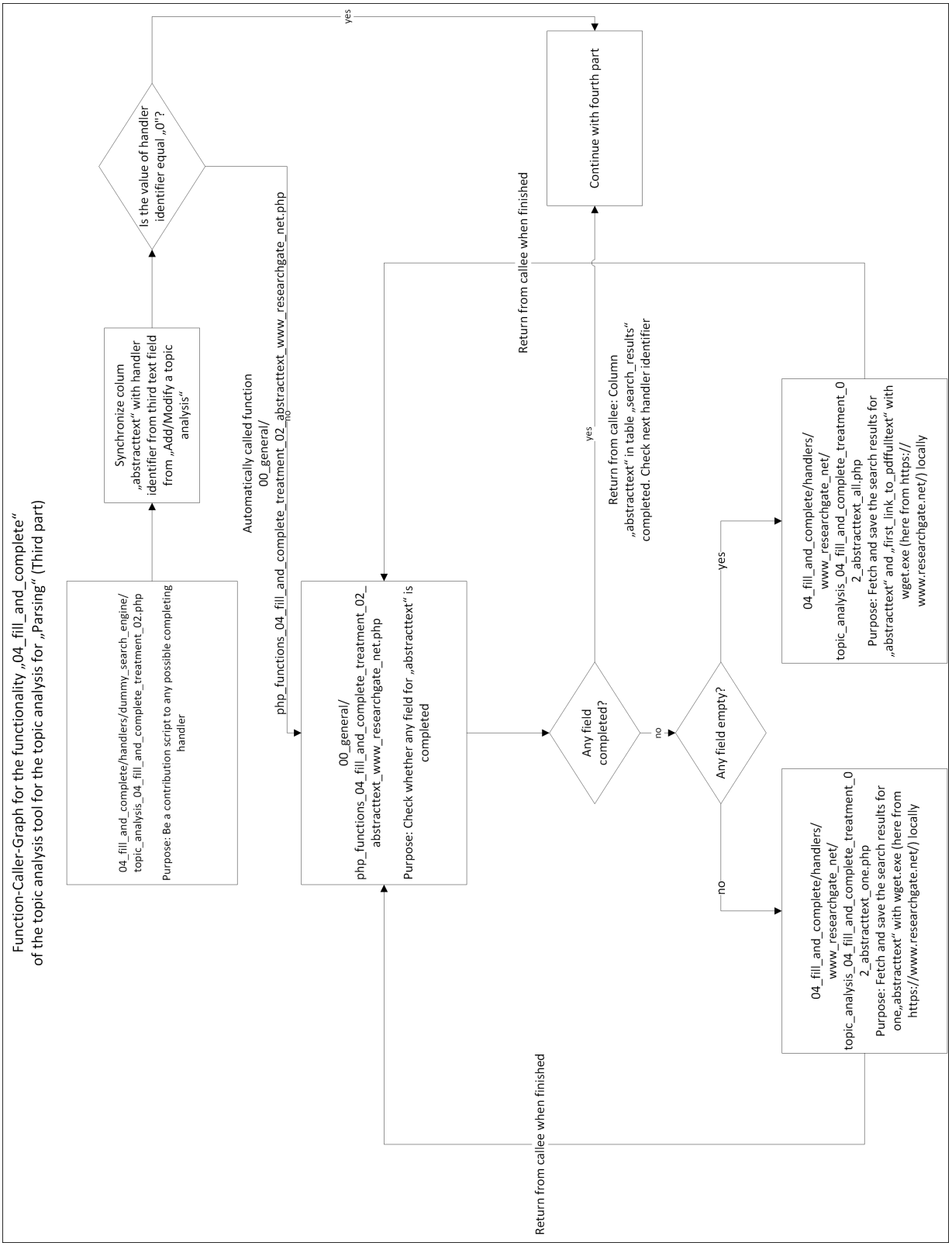


Figure D.4 Fill and complete with the completing-procedure for https://www.researchgate.net/

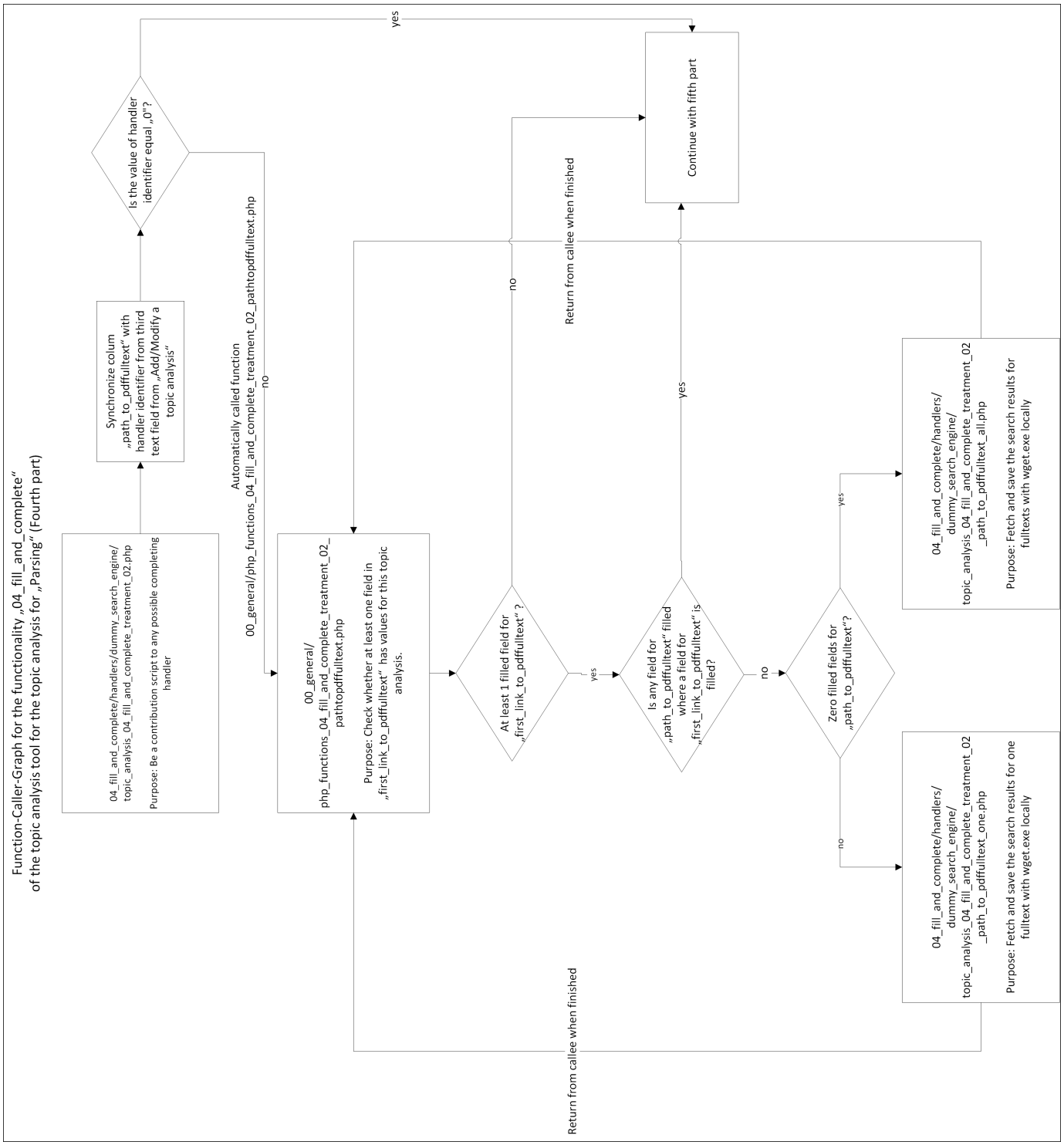


Figure D.5 Fill and complete with the completing-procedure for “path_to_pdffulltext”

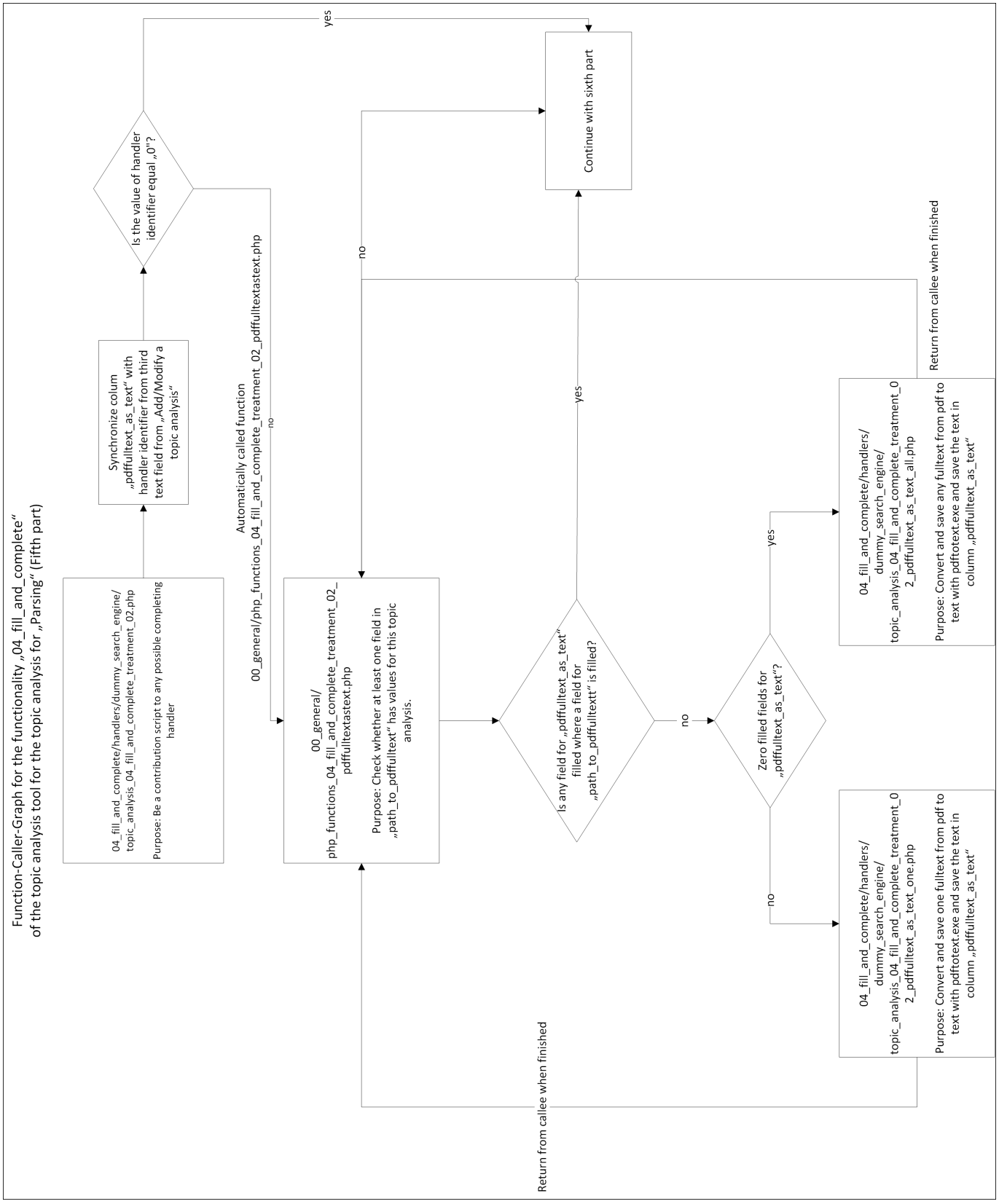


Figure D.6 Fill and complete with the completing-procedure for “pdffulltext_as_text”

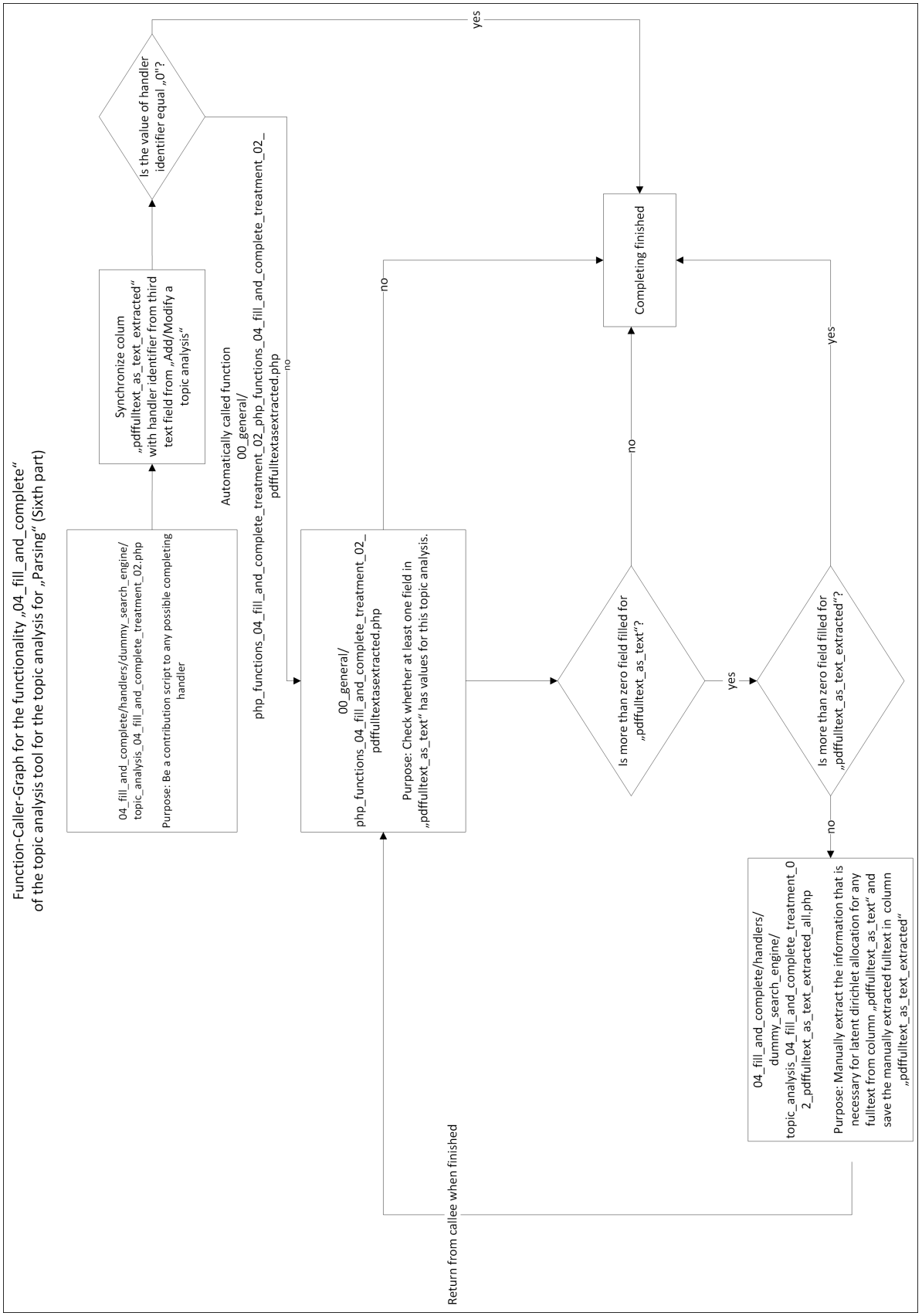


Figure D.7 Fill and complete with the completing-procedure for "pdffulltext_as_text_extracted"

```
$xmltag[0]="<authors>";
$xmltag[1]="</authors>";
$xmltag[2]="<title>";
$xmltag[3]="</title>";
$xmltag[4]="<venue>";
$xmltag[5]="</venue>";
$xmltag[6]="<year>";
$xmltag[7]="</year>";
//Below is the handler for each inputfile that has its input from "http://dblp.org/search/publ/api?q=..."
$firstidentifiercomesagain=true;
$newstrposition=0;
$strposition=0;
$strposition2=0;
$zaehler2=0;
$htmlsource=stringtohtmlstring($htmlsource);
while($firstidentifiercomesagain){
    for($zaehler=0;$zaehler<7;$zaehler++){
        $strposition=stripos($htmlsource,stringtohtmlstring($xmltag[$zaehler]),$strposition);
        $zaehler=$zaehler+1;
        $strposition2=stripos($htmlsource,stringtohtmlstring($xmltag[$zaehler]),$strposition);
        $strposition2=$strposition2+strlen(stringtohtmlstring($xmltag[$zaehler]));
        $strupdatequery[$zaehler2]=excludeanyxmltag(htmlstringtostring(substr($htmlsource,$strposition,$strposition2-$strposition)));
        $zaehler2++;
    }
```

Figure D.8 First part of the filling-handler that handles search results from http://dblp.org/search/publ/api

```
$zaehler2=0;
$returndbconnect=dbconnect();
$returndbinsert=dbinsert($returndbconnect,"search_results","(id_search_strings,authors,title,conference,year)","('$searchstringid'
if($returndbinsert==0){
    echo "INSERT-Query in table search_results for the search string-id= " . $searchstringid . " went wrong.</br>";
}
eliminatefakerowsintablesearchresults();
dbdisconnect($returndbconnect);
//If we do not have a next xml-open-tag equal "<authors>" in the dblp-xml-file we are at the end of the search result
//and we quit the while-loop
$newstrposition=$strposition;
$strposition=stripos($htmlsource,stringtohtmlstring($xmltag[0]),$newstrposition);
if($strposition==false){
    $firstidentifiercomesagain=false;
}
}
                "('$searchstringid', '$strupdatequery[0]','$strupdatequery[1]','$strupdatequery[2]','$strupdatequery[3]')");
```

Figure D.9 Second part of the filling-handler that handles search results from http://dblp.org/search/publ/api

```
if ($file!="." && $file!=".."){
    //we extract the abstract link from the file content
    $returnreadfileindir=htmlstringtostring(read_file_in_dir($dirname,$file));
    $importantbeginofreturnreadfileindir=stripos($returnreadfileindir,'class="publication-title js-publication-title-link');
    $importantmiddle01ofreturnreadfileindir=stripos($returnreadfileindir,'href=',$importantbeginofreturnreadfileindir);
    $importantmiddle02ofreturnreadfileindir=stripos($returnreadfileindir,"",$importantmiddle01ofreturnreadfileindir);
    $importantendofreturnreadfileindir=stripos($returnreadfileindir,'">',$importantmiddle02ofreturnreadfileindir);
    //We do not want " at the end of the link
    $importantendofreturnreadfileindir=$importantendofreturnreadfileindir-1;
    $returnreadfileindir=substr($returnreadfileindir,$importantmiddle02ofreturnreadfileindir+1,$importantendofreturnreadfileindir-$imp
    //and update the field from column first_link_to_abstracttext from table "search_results" where the title's field in this row
    //has the same content as the content of $databaseentry
    $returnupdate=dbupdate($returndbconnect,"search_results","first_link_to_abstracttext='$returnreadfileindir'","id='$id'");
    if($returnreadfileindir==''){
        $returnvalue[0]=false;
        $returnvalue[1]='Error while updating column "first_link_to_abstracttext" in table "search_results: A link that should be upda
    }elseif($id==''){
        $returnvalue[0]=false;
        $returnvalue[1]='Error while updating column "first_link_to_abstracttext" in table "search_results: One or more than one searc
    }elseif($returnupdate==false){
        $returnvalue[0]=false;
        $returnvalue[1]='Error while updating column "first_link_to_abstracttext" in table "search_results: MySQL update query failed.
    }
```

```
        $importantendofreturnreadfileindir-$importantmiddle02ofreturnreadfileindir);
        here the title's field in this row

        rnreadfileindir'","id='$id'");

        rch_results: A link that should be updated to the column "first_link_to_abstracttext" was empty';

        rch_results: One or more than one search result ids were empty.';

        rch_results: MySQL update query failed.';
```

Figure D.10 Completing-handler that handles search results from https://www.researchgate.net/search?q=[$title]

Appendix E

Demonstration of the topic analysis tool for the category “Parsing” as a running example

E.1 lda.R for “CC”, an example latent dirichlet allocation environment for “Parsing”

```
#This program is created with the help of code snippets from
#"https://gist.github.com/not-for-me/f0e269015e5681ec56ab"
#and Carson Sievert, "A topic model for movie reviews" at
https://ldavis.cpsievert.me/reviews/reviews.html at 2017-09-30
library(tm)
stop_words <- stopwords("SMART")
path<-
"C:/xampp/htdocs/topic_analysis/13_execute_LDA/lda_input_execute_ou
tput/Parsing/CC/input/"
mytextcorpus<-Corpus(DirSource(path),readerControl = list(reader=
readPlain,language="en"))
summary(mytextcorpus)

# pre-processing:
mytextcorpus <- gsub("'", "", mytextcorpus)  # remove apostrophes
mytextcorpus <- gsub("[[:punct:]]", " ", mytextcorpus)  # replace
punctuation with space
mytextcorpus <- gsub("[[:cntrl:]]", " ", mytextcorpus)  # replace
control characters with space
mytextcorpus <- gsub("^[[:space:]]+", "", mytextcorpus) # remove
whitespace at beginning of documents
mytextcorpus <- gsub("[[:space:]]+$", "", mytextcorpus) # remove
whitespace at end of documents
mytextcorpus <- tolower(mytextcorpus)  # force to lowercase

# tokenize on space and output as a list:
doc.list <- strsplit(mytextcorpus, "[[:space:]]+")

# compute the table of terms:
term.table <- table(unlist(doc.list))
term.table <- sort(term.table, decreasing = TRUE)

# remove terms that are stop words or occur fewer than 5 times:
del <- names(term.table) %in% stop_words | term.table < 5
term.table <- term.table[!del]
vocab <- names(term.table)

# now put the documents into the format required by the lda
package:
get.terms <- function(x) {
  index <- match(x, vocab)
  index <- index[!is.na(index)]
  rbind(as.integer(index - 1), as.integer(rep(1, length(index))))
}
documents <- lapply(doc.list, get.terms)
# Compute some statistics related to the data set:
D <- length(documents)  # number of documents (2,000)
W <- length(vocab)  # number of terms in the vocab (14,568)
doc.length <- sapply(documents, function(x) sum(x[2, ]))  #
```

```
number of tokens per document [312, 288, 170, 436, 291, ...]
N <- sum(doc.length)  # total number of tokens in the data
(546,827)
term.frequency <- as.integer(term.table)  # frequencies of terms
in the corpus [8939, 5544, 2411, 2410, 2143, ...]

# MCMC and model tuning parameters:
K <-3
G <- 5000
alpha <- 0.02
eta <- 0.02

# Fit the model:
library(lda)
set.seed(357)
fit <- lda.collapsed.gibbs.sampler(documents = documents, K =3,
vocab = vocab,
                                   num.iterations = G, alpha =
                                   alpha,
                                   eta = eta, initial = NULL,
                                   burnin = 0,
                                   compute.log.likelihood = TRUE)

theta <- t(apply(fit$document_sums + alpha, 2, function(x) x/sum(x
)))

phi <- t(apply(t(fit$topics) + eta, 2, function(x) x/sum(x)))

mreviews <- list(phi = phi,
                     theta = theta,
                     doc.length = doc.length,
                     vocab = vocab,
                     term.frequency = term.frequency)
library(LDAvis)

# create the JSON object to feed the visualization:
json <- createJSON(phi = mreviews$phi,
                   theta = mreviews$theta,
                   doc.length = mreviews$doc.length,
                   vocab = mreviews$vocab,
                   term.frequency = mreviews$term.frequency)
serVis(json, out.dir =
"C:/xampp/htdocs/topic_analysis/13_execute_LDA/lda_input_execute_ou
tput/Parsing/CC/output/vis/", open.browser = FALSE)
```

Bibliography

[1] Charu C. Aggarwal. *Data Mining: The Textbook*. Springer Publishing Company, Incorporated, 2015. ISBN: 3319141414, 9783319141411.

[2] Sethi Aho Lam and Ullman. *Compilers: Principles, Techniques, and Tools*. Boston, MA, USA: Pearson Education, Inc., 2007. ISBN: 0-321-49169-6.

[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation". In: *J. Mach. Learn. Res.* 3 (Mar. 2003), pp. 993–1022. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=944919.944937.

[4] Carson Sievert. *A topic model for movie reviews*. https://ldavis.cpsievert.me/reviews/reviews.html. Aug. 2017.

[5] Marcel Heinz. "Systematic Mapping Studies". https://userpages.uni-koblenz.de/~laemmel/esecourse/slides/sms.pdf. Mar. 2017.

[6] Jonathan Chang. *Package 'lda'*. https://cran.r-project.org/web/packages/lda/lda.pdf. Sept. 2017.

[7] Kenny Shirley. *Demo page for LDAvis: A method for visualizing and interpreting topics*. http://kennyshirley.com/LDAvis/. Aug. 2017.

[8] Kenton W. Murray. *Summarization by Latent Dirichlet Allocation: Superior Sentence Extraction through Topic Modeling*. http://www.kentonmurray.com/thesis/thesis.pdf. Sept. 2017.

[9] Microsoft. *Latent Dirichlet Allocation*. https://msdn.microsoft.com/en-us/library/mt762914. Sept. 2017.

[10] No author provided. *$_GET*. http://php.net/manual/en/reserved.variables.get.php. Sept. 2017.

[11] No author provided. *$_POST*. http://php.net/manual/en/reserved.variables.post.php. Sept. 2017.

[12] No author provided. *topicmodel_with_r*. https://gist.github.com/not-for-me/f0e269015e5681ec56ab. Sept. 2017.

[13] Wikipedia Community. *Application programming interface*. https://en.wikipedia.org/wiki/Application_programming_interface. Aug. 2017.

[14] Wikipedia Community. *Information retrieval*. https://en.wikipedia.org/wiki/Information_retrieval. Aug. 2017.

[15] Wikipedia Community. *Paywall*. https://en.wikipedia.org/wiki/Paywall. Aug. 2017.