2020 24th International Conference Information Visualisation (IV) | 978-1-7281-9134-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/IV51561.2020.00065

Comparison of Full-text Articles and Abstracts for Visual Trend Analytics through Natural Language Processing

1st Kawa Nazemi
Human-Computer Interaction & Visual Analytics
Darmstadt University of Applied Sciences
Darmstadt, Germany
kawa.nazemi@h-da.de
ORCID: 0000-0002-2907-2740

2nd Maike J. Klepsch
Human-Computer Interaction & Visual Analytics
Darmstadt University of Applied Sciences
Darmstadt, Germany
maike.j.klepsch@stud.h-da.de
ORCID: 0000-0003-2666-2056

3rd Dirk Burkhardt
Human-Computer Interaction & Visual Analytics
Darmstadt University of Applied Sciences
Darmstadt, Germany
dirk.burkhardt@h-da.de
ORCID: 0000-0002-6507-7899

4th Lukas Kaupp
Human-Computer Interaction & Visual Analytics
Darmstadt University of Applied Sciences
Darmstadt, Germany
lukas.kaupp@h-da.de
ORCID: 0000-0001-9411-6781

***Abstract*—Scientific publications are an essential resource for detecting emerging trends and innovations in a very early stage, by far earlier than patents may allow. Thereby Visual Analytics systems enable a deep analysis by applying commonly unsupervised machine learning methods and investigating a mass amount of data. A main question from the Visual Analytics viewpoint in this context is, do abstracts of scientific publications provide a similar analysis capability compared to their corresponding full-texts? This would allow to extract a mass amount of text documents in a much faster manner. We compare in this paper the topic extraction methods LSI and LDA by using full text articles and their corresponding abstracts to obtain which method and which data are better suited for a Visual Analytics system for Technology and Corporate Foresight. Based on a easy replicable natural language processing approach, we further investigate the impact of lemmatization for LDA and LSI. The comparison will be performed qualitative and quantitative to gather both, the human perception in visual systems and coherence values. Based on an application scenario a visual trend analytics system illustrates the outcomes.**

***Index Terms*—Visual Analytics, Data Science, Natural Language Processing, Visual Trend Analytics.**

I. Introduction

Scientific publications are an essential resource for detecting and predicting emerging technologies and innovations that could strengthen the potentials of economy and society. In particular, approaches from Visual Analytics may lead to detect such emerging technological trends in an early stage and enlighten potential future directions for strategic decision making. Today's Visual Analytics approaches commonly make use of patents' metadata and in some rare cases of scientific publications. Thereby different machine learning methods are applied to extract terms and topics to set them in correlation to the temporal dimension. This information enables detecting emerging or decreasing technological trends and deciding about future directions of technologies and innovative approaches in the related domains. While extracting information from patent data does not really enable an early detection of such trends, scientific publications can reveal technological trends in an early stage and illustrate the continuous temporal spread. Furthermore, scientific publications provide a more recent and up-to-date investigation of the propagated innovations in contrast to patents, due to their shorter publication process. The main challenge here is to extract that kind of information out of the mass amount of scientific publications that increase with an enormous velocity. With today's computational power the mining of information itself is not a real challenge anymore. It is far more the human-centric approach of Visual Analytics. Humans are investigating the visual representations of data and discover new insights out of the interactive visualizations. A main research question in this context is, whether the abstract has the same value as their corresponding full-texts. Extracting information out of hundreds of millions of abstracts is much easier than extracting the information out of their corresponding full-texts. Very few publications to date have investigated this question with contrary results. Most of those comparisons are studies in the domain of Biology, Medicine and Chemistry (BMC) or Biomedical text Mining [1]–[4]. To our best of knowledge there is no investigation of a comparison of *Latent Semantic Indexing* (LSI) and *Latent Dirichlet Allocation* (LDA) in this context. We further investigate both methods with and without lemmatization that could not be found in the literature review as well. Further, we investigate if there are any significant

2375-0138/20/$31.00 ©2020 IEEE
DOI 10.1109/IV51561.2020.00065

differences between some areas of Computer and Information Science. We therefore have chosen three different areas that should illustrate the differences. For this, we will introduce a simple and fully replicable natural language (NLP) process. Our main focus lies on human-centric Visual Analytics for Technology and Corporate Foresight. We therefore, investigate the results in a qualitative manner based on the visual outcomes and on quantitative manner by measuring the coherence values. We start with an investigation of the literature review followed by the model that can easily be replicated to validate the outcomes. Thereafter, the visual representation and the results will be presented and discussed. The paper concludes with an application scenario of Visual Trend Analytics to illustrate how the visual comparison effects the analysis process. We used for our comparison the Springer database and chose three different domains from Computer and Information Science. Thereby overall 2,670 full-text documents and their corresponding abstracts were investigated. Our main contributions are (1) a comparison of LSI and LDA for full-text articles and their corresponding abstracts, (2) a comparison with and without lemmatization and in three different areas of Computer and Informations Science and (3) the application of extracted topics in an Visual Analytics system for gathering insights of emerging trends.

II. Related Work

Samuel et al. [5] compared abstracts and full-texts of about 20,000 articles from PubMed Central (PMC) and the Directory of Open Access Journals (DOAJ) with a special focus on protein-protein interactions mainly through named entity recognition (NER). They found out that explicit protein-protein interactions are only mentioned in the full-texts. Cohen et al. [1] used the *CRAFT* corpus [4] and a process of data cleaning, part-of-speech tagging and named entity recognition to extract information from full-texts and abstracts. They examined structural aspects, e.g. distribution of sentence length and morphosyntactic and discourse features, e.g. incidence of coordination, negation, passives, distribution of semantic classes of named entities, gene/protein names, mutation, drug names and deceases. They found out that part-of-speech taggers perform notably better on abstracts [1, p. 9] and suggested the ability to deal with parenthesized text in full-texts for an improved information extraction. Müller et al. [6] proposed with *Textpresso* an ontology and ontology-based system that facilitates searches of biological entities. Their system made use of 3,800 publications with focus on `Caenorhabditis elegans`. They stated that the recall increased from about 45% to about 75% when including the full-texts. Based on Textpresso, Garten and Altman [7] developed the text mining tool *Pharmspresso*. Pharmspresso is used to support the identification of important pharmacological facts in full-text articles. They used a corpus consisting of 1,025 full-text articles from 343 different journals and proposed that some pharmacological associations can only be found in the full-text. Shah et al. [8] analyzed a data-set consisting of 104 journal articles from *Nature Genetics*, all of which had the following structure: Abstract, introduction, methods, results and discussion. One question was whether the information in the full-text is sufficiently organized so that keywords can be extracted or whether a word has a different meaning depending on the section in which it is located. They found that the rest of the article besides the abstract also contains essential biological relevant information, although the context has to be considered. Lin [2] used the *TREC 2007* [9] with 192,259 full-text articles and the corresponding MEDLINE records as supplementary data to extract data from full-texts and their abstracts. He stated that disregarding syntax, semantics and even word order has proven to be effective in practice. Lin applied two different retrieval methods, the Okapi *bm25* [10] and a modified *tf-idf* retrieval method to extract terms and topics. He found out that searching full-texts is more effective as measured by MAP, P20, and IPR50, especially when spans of the full-text articles are investigated in the text mining process. The main aspect here is the dedicated investigation of spans that allow more precise results. Syed and Spruit [11] applied the Latent Dirichlet Allocation (LDA) [12] to extract topics from full-texts and their corresponding abstracts. They used two data-sets from the domain fishery. One data-set contained 4,417 articles from a single journal and the second one 14,004 articles, where the majority of the first data-set was included in the second one. Besides using regular expressions for gathering the abstracts and a stop-word list with 153 entries, no lemmatization or stemming was used. They state that stemming could result in unrecognizable words and according to [3] does not allow to deduce if a stemmed word comes from a verb or a noun. They used the *gensim* LDA model [13] to generate various number of topics. Therefore, they changed the K parameter (number of topics) from 1 to 40 and used the first 15 words for their evaluation. This was the first evaluation where humans ranked topics in the documents. Their outcomes illustrated that the topics generated from full-texts showed a higher coherence than for abstracts, beside two LDA-models (changed K parameter). But the second and greater data-set showed no significant differences between the topics extracted from abstracts and those extracted from full-texts. Westergaard et al. [14] analyzed a corpus of 15 million scientific full-text and their corresponding abstracts using the PubMed Central and TDM data using NER and rule-based text-cleaning methods. They evaluated their results through sensitivity (true positive rate) and specificity (true negative rate) to detect protein-protein interactions, disease-gene, and protein subcellular associations. They stated that text-mining of full-text articles consistently outperforms using abstracts only.

The literature review illustrates that the comparisons between text-mining of full-text articles and abstracts are mostly performed within the BioMed domain. Only a few works investigated a broader range of domains. None of the above described comparisons investigated how terms or topics may influence Visual Analytics and the human interactive visual information processing.

III. Model for Term and Topic Extraction

In this section we introduce our general model that was applied to extract terms and topics for the comparison of scientific full-texts and abstracts. We used for our comparison data from the "Springer" data-base through their open API [15].

Overall 2.670 documents were gathered through the application interface of Springer that should fulfill the following criteria:

- the documents should primarily investigate topics concerning Computer and Information Science
- the documents should cover different areas of Computer and Information Science to investigate if the subtopic is of relevance or not
- the documents should be available as open access to ensure a comparison between full text articles and their corresponding abstracts through LDA and LSI

As the literature review revealed, most comparisons were performed in the domain of BioMed. To investigate the effect on other domains, we defined the first criterion. To meet this, we defined three queries that are commonly used in the domains of Computer and Information Science, the brackets should allow to refer to the different queries in the following sections and should be seen as abbreviations: 1: "digital library" OR "digital libraries" (*digital library*) with , 2: "web analytics" OR "digital analytics" OR "information visualization" OR "information visualisation" (*digital analytics*) and 3: "unsupervised learning" (*unsupervised learning*).

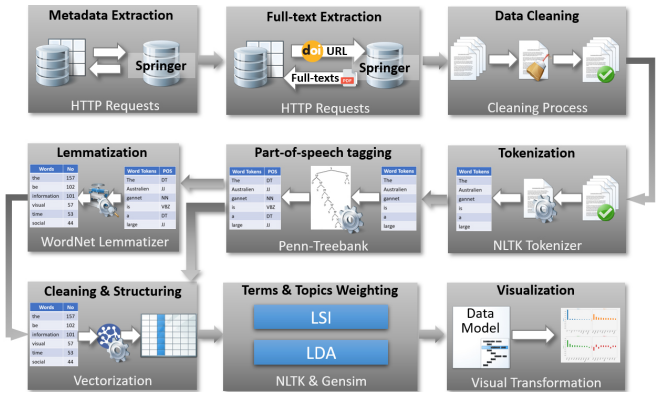

Fig. 1. Natural Language Processing Model for term and topic extraction through an application programming interface.

The above queries were chosen to gather a different thematic areas as the second criterion states. We therefore searched through the Springer API with different queries and refined these to get at all documents available as open access. Each query was further supplemented with the logical statement "AND openaccess:true", e.g. the for *digital library* the query to the API was set as follows: '*"digital library" OR "digital libraries" AND openaccess:true*'. With this statement the third criterion was fulfilled too. All results were open access articles in the Springer database. Overall, the number of the gathered results were as follows, whereas the cleaning and filter process reduced the amount publications slightly: (1) *unsupervised learning*: 1,390 publications (reduced to 1,309), *digital library*: 893 publications (reduced to 839), and *digital analytics*: 541 publications (reduced to 422). This data are the baseline of our comparison. To perform a comparison between full-text articles and their corresponding abstract we set up a model that can be used for further purposes too. Our model consists of six steps beginning with Metadata Extraction up to Visualization as illustrated in Figure 1.

A. Metadata Extraction

In the first step of our model we gather the metadata of the above described queries through the application programming interfaces of Springer [15]. We used for this step the "Springer Nature Metadata API" and the "Springer Nature Open Access API" through http-request. The result set already contains the abstract of each article in most cases, regardless of whether it was open access or not and of the Document Object Identifier (DOI) of each article. Each "RESTful" request has to contain the collection that identifies the repository, e.g. "metadata" for the metadata collection and the result format, e.g. pam or json. The amount of the result set was constrained to 50 entries during our study for each query, so that the requests were performed in several iterations to get all articles for a certain query. In this first step, the collected metadata for all queries was saved into a database with the DOI as identifier. In some cases the same articles were gathered through iterative processing. In these cases the DOI was used to identify, if the metadata for a certain article is already stored to avoid any duplicates in the database containing the metadata. We further integrated a kind of filter that enabled us to decide how many pages an article should contain to be part of our data-set. Thus, we are investigating in particular the difference between full-text articles and abstracts, we decided to investigate only articles that are longer than four pages.

With this first step of our model, we were able to gather the most relevant information from the database, extract most of the abstracts and store the gathered information with a unique identifier.

B. Full-text Extraction

The second step of our model gathers the corresponding full-text articles based on the queried results and through the unique Document Object Identifier (DOI). We gather all the full-text documents through the "Springer Nature Open Access API" with the corresponding DOI. Therefore each document has to be gathered separately with a request containing the DOI. The http-request containing the DOI returns a single article as a PDF-document. This document has to be converted into plain text to process the further steps. We therefore use the *pdfminer* [16] to convert the PDF-document into a text document. After having the text as string, we first check, whether the abstract of this document has already been returned with metadata request. Thus, a number of results do not contain the abstract, the first step is to gather the abstract.

Therefore, we search the string for the term “abstract” starting with the top of the document. Thereafter, line breaks and some terms are searched for identifying the abstract. In most cases the terms “keywords” and “keyword” indicates the end of an abstract, whereas in some cases there are no keywords defined and terms like “introduction” indicates the end of the abstract.

We could observe that older articles does not necessarily contain abstracts or any other kind of text-structuring. These articles just highlight the title and the author in the PDF-document with bold font. There is no abstract, no introduction and no keywords. These articles are omitted, due to the fact that a comparison between full-text and abstract is not possible. These results also reduce the amount of articles. With this second step all full-texts and the missing abstracts are collected and stored in the database for further processing. The extracted abstracts and full-text documents that are stored as text files contain additional information that may lead to wrong or not precise information.

C. Data Cleaning

The third step of our model cleans the extracted data from unnecessary information fragments that could lead to not precise information in the term extraction step. The abstracts were cleaned first. To find out how to clean the abstracts, we investigated a set of 150 documents manually. We could find out that the abstracts commonly start with the word “Abstract” without a space between the word and the real abstract. Further, we could determine that abstracts may start with “AbstractBackground” or “A bstract”. Therefore the first step is removing these words from the abstract. Thus, these terms are directly connected with the first word of the abstract, the words are searched and cut from the abstract itself. Some abstracts further contain an environment starting with “Background” followed by “Results” and “Conclusions”, we also remove these words, since these words may lead to wrong term extraction. We further observed that abstracts gathered from the full-texts as described in Section III-B may contain hyphens at the end of the line. We therefore removed these through the simple function *removeHyphen* (see Listing 1).

```
def removeHyphen(textfile):
    newText = ''
    for line in textfile:
        if (line.endswith('-\n')):
            newText += line.rstrip('-\n')
        else:
            newText += line + '\n'
    return newText
```

Listing 1. Removing the hyphens.

Cleaning the full-text documents requires a bit more steps, due to the different information that an article may contain.

We first removed the hyphens just as described in Listing 1. Thereafter, we identified any numbers and information above the title. Therefore some rules were implemented based on our investigation of more than 100 articles, e.g.:

- Search for the term “Open Access”. If this term is given in the text and the lines above are not more than six, cut the text and lines before.
- Search for different terms that are capitalized, e.g. “BIOMEDICAL ENGINEERING SOCIETY” or “ORIGINAL RESEARCH”. If such terms appear and the lines above are not more than six, cut the the text and lines before.
- Search for the term “Check for updates”. If this term appears in the text and the lines above are not more than six, cut the the text and lines before.

Commonly the second rule with capitalized words led to the effect that unnecessary information above title was deleted. After cutting the information above the title, some similar rules were adapted to delete the authors and everything else that is still listed between title and “Introduction”. Authors’ names could be easily identified, since this information is included in the metadata of the corresponding article. Since the abstracts are still contained in the existing full-texts, it was necessary to remove the abstracts in order to avoid redundant information that could lead to falsified results. Depending on the structure of the text, an abstract can be filtered out by using regular expressions. The abstract is usually introduced by the word “Abstract”. This can be used to determine where the abstract begins. It then depends on which structural parts the abstract ends with. In many cases, the abstract environment ends with the listing of keywords. It is therefore conceivable to remove the content between the terms “Abstract” and “Keywords” using regular expressions. However, this does not apply to all texts since some texts do not contain keywords. For these cases we observed a couple of hundred of documents and defined further regular expressions, e.g. all terms between the terms “Abstract” and “Introduction”. Thereafter we applied regular expressions to delete hyperlinks, emails, symbols and numbers from the entire text. The references were cut by searching for the terms “References”, “Acknowledgments”, “Acknowledgment”, and “Online Supplementary Material”. The texts were searched starting from the last page. If one of the above terms appeared, the search was continued to find another one and if the last term was found, the term and the text below were removed. We applied this rule based on the fact that the acknowledgments are commonly above the references.

D. Tokenization

For the further processing of the cleaned documents, we applied lexical tokenization on word-level to the entire full-text documents and abstracts. For this, we used the predefined function *word_tokenize* of *NLTK* [17] that creates tokens of words for the entire string that is given to the function. Additionally, we used regular expressions to remove numbers and special characters. The application is pretty simple and reliable.

E. Part of Speech Tagging

In the next step of our model, we apply grammatical tagging with part-of-speech-tagging (POS-tagging). For this purpose, we use the *NLTK* [17] tag-set and the corresponding function *nltk.pos_tag*. This step is just another preprocessing step to get the word types contained in the abstracts and

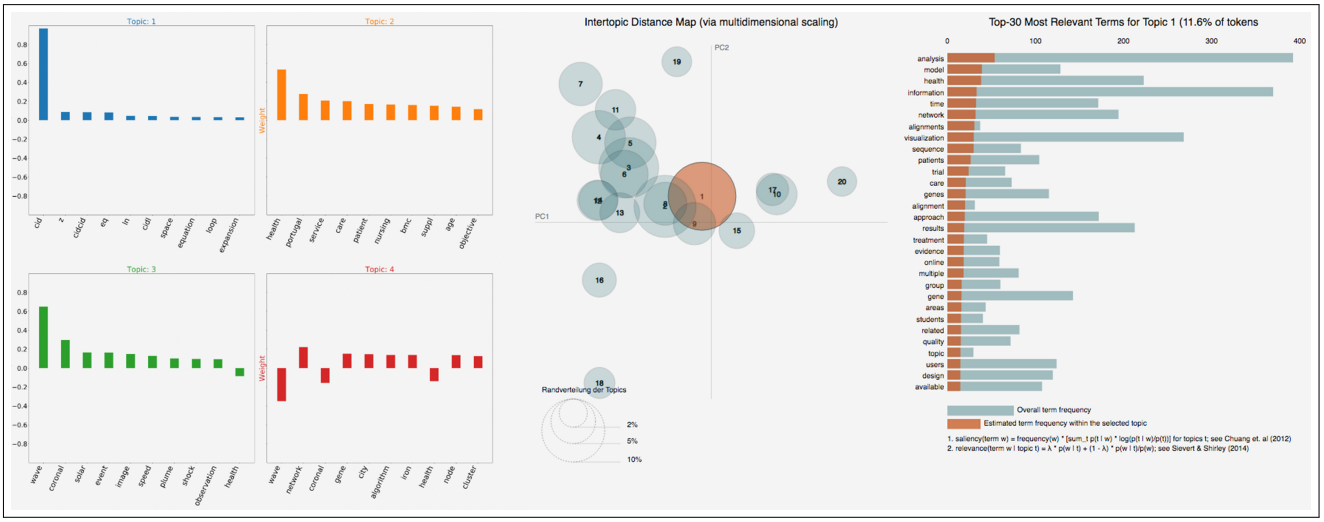

Fig. 2. A first visualization of the results for identifying the outcomes based on humans' perception

full-text articles to stem the words in the next step. The *NLTK pos_tag* is a simple and reliable function that returns a tag containing the information about each word. It uses a special "tagging-grammar" that is not always compatible with other lemmatizers from other providers. NLTK uses the *Penn Treebank* [18] tagset that includes a huge set of further information about the words in addition to word-types [18, p. 317].

F. Lemmatization

Due to the fact that we are interested in comparing whether a method with or without lemmatization gives us better results, this step is performed once and skipped once for the same set of data. In the step of lemmatization the words are transformed into their base form, the so called *lemma*. This leads to a reduced amount of words and in particular to lemmatized words that reduce unambiguity and word variations. So, instead of "is" or "were", the word "be" is stored. The previous step of POS-tagging makes it possible to differentiate between verb and noun, which is necessary for a lot of words. That is also why we deliberately chose not to use stemming but lemmatization, since stemming leads to strings of characters which are not words themselves. In the context of this work the lemma of a word is more meaningful than the word stem alone. For this step, we also used the NLTK library [17]. However, the *WordNetLemmatizer* of *nltk.stem* stems the words using a different POS-tagger than the one we chose. Therefore we converted the tags according to the notation of *WordNet* ([19], [20], [21]).

After having lemmatized the words, it is necessary to remove "stop words" from all documents (abstracts and full-texts). We used the stop word list from NLTK (*stopwordsEN*) [17] and complemented it with a set of words that we manually categorized as stop words. This set was built through the investigation of publication related terms, e.g. "et", "al", "figure", "fig" and all written out numbers like "three" or "four". The generated stop word list will be provided as open source.

G. Vectorization

We needed to convert both collections of text documents to a matrix of token counts to build the topic models. We did this by using the *CountVectorizer* function from the *scikit-learn* library [22] and train a vocabulary dictionary of term-document matrix through the function fit_transform from the same library. With these two steps of vectorization, we have a document matrix of token counts and a term-document matrix of our entire set of data.

H. Term and Topic Weighting

For extracting and weighting terms and topics, we used two different topic models: the probabilistic generative model *Latent Dirichlet Allocation* (LDA) [12] and *Latent Semantic Indexing* (LSI) [23], which uses Singular Value Decomposition (SVD). To our best of knowledge there is no work so far on comparing abstracts and full-texts using LSI. In total we built four models: LDA with and without lemmatization and LSI with and without lemmatization. We created the models by using the *gensim* library [13]. For applying the LDA algorithm we used the function *ldamodel* with K-values (*num_topics*) of 20, since this is the default value. For the LSI model we also used K=20 and set the decay parameter to 0.5 in the function *lsimodel*. Lastly, we determined the coherence values for all models and save them in a separate JSON file.

I. Visualization

The comparison between the models and the different data (full text articles versus their corresponding abstracts) were performed to identify the best fitting data-set and model for visual trend analytics. In a first step, we visualized the extracting data through a simple bar chart and a simple visualization (*pyLDAvis* [24]) that allowed us to compare the result visually

TABLE I
DETERMINED COHERENCE VALUES OF THE MODELS USED FOR $K = 20$

	Digital Analytics		**Digital Libraries**		**Unsupervised Learning**	
Model	Abstracts	Full-texts	Abstracts	Full-texts	Abstracts	Full-texts
LDA with lemmatization	0.3159	0.3207	0.3142	0.4270	0.3037	0.3892
LDA without lemmatization	0.3170	**0.3763**	0.3004	0.4423	0.3323	**0.4788**
LSI with lemmatization	0.3116	0.3125	0.3023	0.4587	0.3043	0.4355
LSI without lemmatization	0.3224	0.3305	0.3070	**0.5121**	0.3291	0.4021

and get an idea if the important information of the data could be visualized for analyzing trends (see Figure 2). This was a first attempt to "see" the results before including them into a real world application. This visual comparison allowed us to gather some major information:

- main technologies that could be used for technology foresight in Visual Analytics could be identified better through the abstracts
- The outcomes from LDA without lemmatization could gather more technological entities for Corporate Foresight
- the difference between LDA and LSI was significantly high. LDA has gathered other topics than LSI. A deeper investigation is necessary due to this outcome.

However, the visual results are not valid for a deeper investigation. In the following section the results will be discussed based on topic coherence measurements [25]. Further the outcomes are visualized in a real world system that integrates a greater data-set in the Section IV to illustrate the main idea of the topic extraction methods.

J. Results

Since topic models usually do not guarantee ideal interpretability, we determined coherence values. The underlying idea of topic coherence goes back to the distribution hypothesis of Harris [25], which states that words with similar meanings often occur in similar contexts. Table I shows the determined coherence values for the individual models and for each topic area. The coherence values of this table thus provide a measure for assessing the quality of the learned topics of a model. Thereby the highest coherence value is highlighted as bold. The results show that

- chosen areas show that a different model is more appropriate, so LDA for unsupervised learning and digital analytics and LSI for digital libraries
- the coherence value of abstracts shows no significant differences and are all about 0.30
- in all cases a significant higher coherence value was achieved for the full-text data compared to the corresponding abstracts
- the models without lemmatization predominantly show a higher coherence value, whereas these values are not statistically significant in all cases. However, this may be due to the fact that the terms within a topic, which are derived from the same lemma, are more similar to each other and thus increase the coherence value accordingly.

We first assumed that the LDA provides a higher coherence value because of the amount of data entities, but digital analytics has the lowest number of entities. The results further showed in a qualitative investigation that the "core" topics of a publication is gathered better through the abstracts, thus these illustrates the main ambition of a publication. Full text articles may contain a huge number terms that are not really related to the core of a scientific publication and may not provide technologies or strategies in strategic analytical systems. Our investigation further showed that our enhancements on the stop-word list should be further complemented, thus in full-texts terms like "journal", "page", "article", and in abstracts terms like "paper" or "result" appeared very often that may have led to less accurate results.

IV. APPLICATION SCENARIO

We investigated a coherence value measurement according to Harris [25] and a visual comparison of the extracted terms by LSI and LDA for a Visual Analytics system for Corporate Foresight. In such a system the human's visual perception plays an essential role [26] that could lead to perceive, interpret and analyze the extracted information much faster.

According to the coherence values and the qualitative investigation, we applied the non-lemmatized LDA topics to a Visual Analytics system for detecting and predicting emerging technologies and innovations from scientific publication. For this purpose, we used the entire data-set of the "Springer" database with overall more than 22 million entries and extracted through the DOI the corresponding abstracts. Furthermore, the "DBLP" metadata [27] and the data of the EuroGraphics Association are partially integrated. Through data enrichment and modeling methods a variety of data-models were created that allowed us to detect emerging trends, explore author relationships, identify countries, affiliations and authors in certain domains of interest and include a variety interaction and faceting methods [28].

The integrated data models allow us to provide several interactive visual layouts that enable information gathering from different perspectives [29]. An overview of emerging topics gathered from the entire database is illustrated in Figure 3-a. This overview on macro-level provide a first overview of all emerging topics in a certain database [30].

For analyzing trends, it is important to gather the knowledge of the underlying topics, technologies etc. emerged during the time or lost its relevance. Figures 3-b and 3-c illustrate two visual layouts that make use of the different data models. To

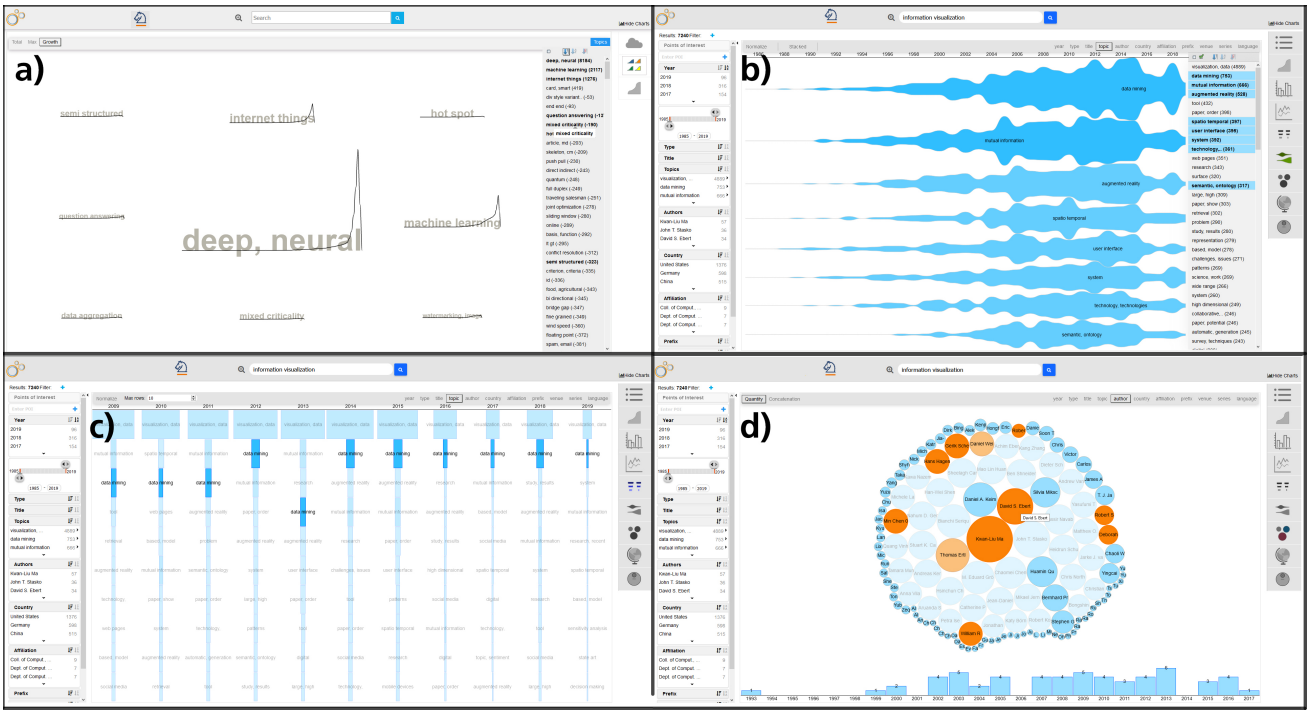

Fig. 3. A: Emerging trends extracted from topics in correlation to time; B: temporal spread of topics; C: a different perspective on topics and D: coauthors and temporal view of the publications of a certain author.

enable a fast and comprehensible analysis, we integrated a temporal ranking (see Figure 3-c). This visual layout offers not only a number of configuration areas, but also the ability to specify the amount of rows to be visualized. The visual layout is divided horizontally into columns for each year of the analyzed time span. The arrangement is based on the amount of publications having a topic or another selected facet item as a property of the selected facet type, sorted in descending order from top to bottom. The order only represents the ranking, additional more concrete information about the relative amount is represented by the width of each rectangle. With these position and form indicators, the user can quickly determine topics and terms with high influences for each year. In Figure 3-b a temporal data model, a topic model and a semantic model are merged to get the visualized information. Thereby, top related topics of documents from the result set of the query "data mining" are visualized and by selecting one topic, the temporal ranking for each year is highlighted by user's selection. In Figure 3-d a semantic visual layout visualizes the authors' correlations (co-author information) and between topics (topic correlations) and the semantic correlation between the information entries. Commonly semantic relations are visualized with node-link graphs, which may lead to complex visualizations and reduce the analysis capability. We integrated beside such node-link visualizations a circle-layout that arranges the entities as a spiral starting from the center of the screen. The viszalization illustrates the semantic relations based on the facet type author (co-author information). The size of each element indicates the amount of publication per author, whereas the degree option indicates the amount of distinct relation targets within the facet type. The Semantic Visual Layout is used to provide detailed relational information about individual facet items, which can be accessed through user interaction. After selecting a circle all relational information within the same facet type are highlighted. This leads to a real-time loading of all co-authors in the result set. Further, users are able to get an insight about correlations within the semantic relations through mouse-over that refines the co-authorship of certain authors through color. Further integrated viszualizations illustrate a topic visual layout that visualizes topics related to search term based on the frequency of their appearance, and geographical visual layout that encodes the amount of publications per authors' country through saturation.

V. Conclusion

We conducted in this paper a comparison of LSI and LDA with and without lemmatization for three areas of Computer and Information Science with the goal to process data for a visual trend analytics system. We started with an investigation of the literature review and could outline that such a comparison is not provided neither for Visual Analytics systems nor for comparing LSI and LDA with and without lemmatization. Thereafter, we introduced a natural language processing approach to extract terms and topics from text and illustrated in detail the procedure with naming all the libraries

that were used. We tried to keep the approach as simple and replicable as possible by using standard libraries describing all steps in a replicable way. The comparison was conducted with 2,670 full-text documents and their corresponding abstracts from the Springer Metadata API and the Springer Nature open access API. Thus the comparison was performed for a Visual Analytics system, we provided first a visual comparison based on the way how Visual Analytics systems are used by humans. The visual comparison illustrated that the main technological aspects of a paper are already included in the abstracts. The visual comparison was performed qualitatively. Thereafter, we illustrated the coherence values based on Harris coherence value measurements [25]. Our results illustrate that LSI and LDA differ even within the domain of Computer and Information Science based on selected areas. It further illustrated that lemmatization and LDA or LSI leads to lower coherence values. In all cases a higher coherence value was achieved for the full-text data compared to the corresponding abstracts. The results were further visualized in a comparative manner to illustrate based on our application scenario the visual analysis capabilities of the extracted terms combined with further data-models. We concluded our paper with an application scenario for Visual Trend Analytics.

ACKNOWLEDGMENTS

Acknowledgments will be added after peer-review. This work was partially funded by the Hessen State Ministry for Higher Education, Research and the Arts within the program "Forschung für die Praxis" and was conducted within the research group on Human-Computer Interaction and Visual Analytics (https://vis.h-da.de). The presentation of this work was supported by the Research Center for Applied Informatics (FZAI) of the Darmstadt University of Applied Sciences.

REFERENCES

[1] K. B. Cohen, H. L. Johnson, K. Verspoor, C. Roeder, and L. E. Hunter, "The structural and content aspects of abstracts versus bodies of full text journal articles are different," *BMC Bioinformatics*, vol. 11, no. 1, p. 492, Sep 2010. [Online]. Available: https://doi.org/10.1186/1471-2105-11-492

[2] J. Lin, "Is searching full text more effective than searching abstracts?" *BMC Bioinformatics*, vol. 10, no. 1, feb 2009.

[3] N. Evangelopoulos, X. Zhang, and V. R. Prybutok, "Latent semantic analysis: five methodological recommendations," *European Journal of Information Systems*, vol. 21, no. 1, pp. 70–86, Jan 2012. [Online]. Available: https://doi.org/10.1057/ejis.2010.61

[4] K. Verspoor, K. B. Cohen, and L. Hunter, "The textual characteristics of traditional and open access scientific journals are similar," *BMC Bioinformatics*, vol. 10, no. 1, jun 2009.

[5] J. Samuel, X. Yuan, X. Yuan, and B. Walton, "Mining online full-text literature for novel protein interaction discovery," in *2010 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)*, Dec 2010, pp. 277–282.

[6] H.-M. Müller, E. E. Kenny, and P. W. Sternberg, "Textpresso: An ontology-based information retrieval and extraction system for biological literature," *PLOS Biology*, vol. 2, no. 11, 09 2004. [Online]. Available: https://doi.org/10.1371/journal.pbio.0020309

[7] Y. Garten and R. B. Altman, "Pharmspresso: a text mining tool for extraction of pharmacogenomic concepts and relationships from full text," in *BMC bioinformatics*, vol. 10, no. 2. BioMed Central, 2009, p. S6.

[8] P. K. Shah, C. Perez-Iratxeta, P. Bork, and M. A. Andrade, "Information extraction from full text scientific articles: Where are the keywords?" *BMC bioinformatics*, vol. 4, no. 1, 2003.

[9] W. R. Hersh, A. M. Cohen, L. Ruslen, and P. M. Roberts, "TREC 2007 genomics track overview," in *Proceedings of The Sixteenth Text REtrieval Conference, TREC 2007, Gaithersburg, Maryland, USA, November 5-9, 2007*, 2007. [Online]. Available: http://trec.nist.gov/pubs/trec16/papers/GEO.OVERVIEW16.pdf

[10] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, M. Gatford, and A. Payne, "Okapi at TREC-4," in *Proceedings of The Fourth Text REtrieval Conference, TREC 1995, Gaithersburg, Maryland, USA, November 1-3, 1995*, 1995. [Online]. Available: http://trec.nist.gov/pubs/trec4/papers/city.ps.gz

[11] S. Syed and M. Spruit, "Full-text or abstract? examining topic coherence scores using latent dirichlet allocation," in *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, oct 2017.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, 2003. [Online]. Available: http://www.jmlr.org/papers/v3/blei03a.html

[13] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

[14] D. Westergaard, H.-H. Stærfeldt, C. Tønsberg, L. J. Jensen, and S. Brunak, "A comprehensive and quantitative comparison of text-mining in 15 million full-text articles versus their corresponding abstracts," *PLOS Computational Biology*, vol. 14, no. 2, p. e1005962, feb 2018.

[15] S. Nature, "Text and data mining at springer nature," online, 2019, accessed on June 1st 2019. [Online]. Available: https://www.springernature.com/gp/researchers/text-and-data-mining

[16] Y. Shinyama, "Pdfminer.six," online, 2014, accessed: March 2019. [Online]. Available: https://github.com/pdfminer/pdfminer.six

[17] NLTK, "Nltk 3.4.1 documentation," online, 2019, accessed March 2019. [Online]. Available: https://www.nltk.org/api/nltk.tokenize.html

[18] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of english: The penn treebank," *Comput. Linguist.*, vol. 19, no. 2, pp. 313–330, Jun. 1993. [Online]. Available: http://dl.acm.org/citation.cfm?id=972470.972475

[19] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995. [Online]. Available: http://doi.acm.org/10.1145/219717.219748

[20] C. Fellbaum, Ed., *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press, 1998.

[21] WordNet, "Wordnet - a lexical database for english," online, Princeton University, 2010, accessed March 2019. [Online]. Available: https://wordnet.princeton.edu/

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[23] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[24] C. Sievert and K. Shirley, "Ldavis: A method for visualizing and interpreting topics," in *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, 2014, pp. 63–70.

[25] Z. Harris, "Distributional structure," *Word*, vol. 10, 01 1954.

[26] K. Nazemi, *Adaptive Semantics Visualization*. Springer International Publishing, Studies in Computational Intelligence 646, 2016.

[27] M. Ley, "dblp computer science bibliography," online, 2018, accessed June 2019. [Online]. Available: https://dblp.uni-trier.de/

[28] K. Nazemi and D. Burkhardt, "Visual analytics for analyzing technological trends from text," in *2019 23rd International Conference Information Visualisation (IV)*, 2019, pp. 191–200, best Paper Award.

[29] K. Nazemi, R. Retz, D. Burkhardt, A. Kuijper, J. Kohlhammer, and D. W. Fellner, "Visual trend analysis with digital libraries," in *Proceedings of i-KNOW '15*. Graz, Austria: ACM, 2015, pp. 14:1–14:8.

[30] K. Nazemi and D. Burkhardt, "A visual analytics approach for analyzing technological trends in technology and innovation management," in *Advances in Visual Computing*, G. B. et al., Ed. Cham: Springer International Publishing, 2019, pp. 283–294.