Karlsruhe Institute of Technology

Visualization of Topic Models for Patent Documents

BACHELOR THESIS

of

Paolo Ramella-Ratin

Student-ID: 1958685

Study programme: Industrial Engineering and Management

Institute of Economics (ECON)

Chair in Economic Policy

Examiner: Prof. Dr. Ingrid Ott

Supervisor: M. Sc. David Bälz

Time period: 01.05.2019 – 31.10.2019

Ich versichere wahrheitsgemäß, die Arbeit selbstständig verfasst, alle benutzten Quellen und Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde sowie die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben.

Karlsruhe, den

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Research Questions 2

1.3 Relevance 2

1.4 Topic Models 3

1.5 Patent Data 4

2 Literature **6**

2.1 Visualisation 6

2.2 Metrics 9

3 Tool Design **12**

3.1 Requirements 12

3.2 Implementation 13

3.3 Tool Overview 15

4 Conclusion **18**

4.1 Summary 18

4.2 Outlook 19

A Appendix **20**

A.1 visualization.py 20

A.2 utility.py 20

A.3 pmi.py 20

References **21**

1 Introduction

This work is about the visualization of topic models that have been trained on patent documents. Over the course of the first chapter we present a motivation into the subject, our research questions as well as the relevance of the thesis. The reader is also introduced to the subject of topic models and gets an understanding of what patent data consists of. In the following chapter the reader is presented with an overview of visualization techniques as well as available tools and metrics applied to topic models, through a literature overview. In the third chapter we present the implementation of a visualization tool developed alongside the thesis. This work is rounded off by a conclusion and an outlook.

1.1 Motivation

Thanks to the developments in the fields of machine learning and natural language processing we are now able to systematically analyse large corpora. There are many complex and powerful algorithms targeting multiple applications in regards to textual data. But the popular text mining methods grouped under the term of topic models are particularly well suited for understanding a corpus of documents. Topic models operate by statistically discovering which underlying thematic structure is represented in a given corpus. Unfortunately simply applying existing algorithms to a set of data is not enough. It is essential to present and visualize results in a way that is tailored to the topic models field of application. The output of a topic model has to be contextualised, enhanced and curated. In this context visualization serves as a way to communicate, compare and comprehend the work done by data scientists and researchers.

On the other hand we have seen the digital take on central parts of our every day lives. These technologies have disrupted many aspects of our lives trying to simplify or enhance our interactions, business processes and so on. An often overlooked consequence of these changes is the way patent use has shifted. Indeed companies use patents not to secure the technology itself but to create a unique selling point in the eyes of investors and defend their entry into new markets. As described by Hall (2004) the companies linked to those changes are mostly based in the electrical and computing industries. Besides other factors, innovation remains a central driver

of economic growth. As patents allow us a deep look into said innovations they serve us insight into the economy.

1.2 Research Questions

In the context of this bachelor thesis we explore the following different but related research questions:

1) What existing methods and tools for visualizing topic models are present in the literature? We will explain what the techniques consist of and the data they require in order to be implemented. We will describe the tools design choices as well as the way they operate. We will enumerate the challenges and opportunities in using the visualization techniques and tools. What metrics for evaluating topic models and enhancing visualizations are presented in the literature? We will explain how well some metrics perform the evaluation tasks and how they compare to one and another. We will also list the prerequisites and challenges to computing the metrics. These questions will be answered through an overview of the literature available on topic model visualization and the associated metrics.

2) What are the essential design choices for a visualization tool build for topic models trained on patent data? The answers to the previous question will build the basis to answer this one. We will explain what makes a good visualization technique. We also want to describe the metric most appropriate to evaluate and enhance the visualized topics. We will describe how to best link the corpus of patent documents to the visualization tool. These questions will narrow down what makes a good visualization tool in order to succeed the implementation.

3) What is the best visualization tool we can implement? We will describe our choice of modern technologies in the design of this tool. We will explain the practicality of our tool. We want to explain the challenges that occurred during the development process. We also want to lay out the advantages in using our tool. Answering these questions will be the realization of the literature review and design choices.

1.3 Relevance

In order to legitimize this work we want to summarize the relevance of this thesis.

The scientific relevance of this thesis stems from the fact that the field of topic model visualizations is a scarce one. Not much recent work is to be found and even more so on topic models trained with patent data. With this thesis we want to further knowledge in this field.

In regards to the educational relevance, the combination of statistics and computer science being applied to an economic domain positions this thesis right at the core of the industrial engineering and management course. This topic therefore makes for an excellent thesis subject for an industrial engineering student.

The practical relevance results from the tool implemented alongside the academic work of this thesis. If good enough the tool might be used and adapted by researchers of the Chair of Economic Policy at the Karlsruhe Institute of Technology.

Finally this thesis is of personal relevance to me because I enjoy working on new technologies and programming projects in general. I am particularly interested in this project because it consists of a creative programming task resulting in a useful output. I had the pleasure to work on a seminar held by the Chair in Economic Policy before my bachelor thesis. Therefore I am partially familiar with the required technologies and have acquired valuable knowledge in the domain of patents.

In the two following sections the reader is introduced to the two underlying subjects of this thesis. Both sections mean to lay down the basic knowledge to understand the task of visualization of topic models trained on patent documents. In the first section the reader learns about topic models. In the second section we present patent documents and how the associated data is structured.

1.4 Topic Models

The technicalities and inner workings of topic models are not the focus of this work. Understanding in a simplified way how they work is still essential to the reader. This section helps the reader contextualise the work present in the coming chapters. The following explanations also give the reader an understanding of where different parts of a topic model might fit into the visualization.

The explanations we will give in this section are based on the latent Dirichlet allocation (LDA) presented by Blei et al. (2003). It is one of the simpler topic models. Over the years a lot of research on extending the LDA topic model has been done and numerous alternatives have been proposed throughout the literature. LDA is a statistical model which organizes a document corpus into unobserved groups called topics which can be interpreted as the themes of the corpus. Each topic is a probability distribution over all the words of the corpus. Using LDA every document in the corpus is assigned a handful of topics.

The assumption that is central to the LDA model is the generative process whereby a corpus arose. This process represents documents as the products of random variables being drawn amongst

others from a particular distribution, namely the Dirichlet distribution. The particularities of the chosen distributions allow topics to consist of a few highly probable words and documents to consist of only a few topics. This allows the topic model to better reflect the reality of documents which most of the times consist of only a few themes chosen by the writer ahead of or during the writing process.

When training topic models on documents and other textual sources the fundamental process is the posterior inference. This process reverses the generative process and determines the distribution of topics which best explains the observed corpus. There a many different inference algorithms available to researchers but they all essentially assign a topic to each word present in the corpus. This assignment might be different depending on the context the word. The output of this step is a topics and vocabulary matrix where each word is assigned a probability reflecting its contribution to the topic. The LDA model also outputs a documents and topics matrix where each document is assigned topics it consists of through weighted scores. Both matrices form the standard output of a topic model. In practice when training a topic model users can easily add other objects to the output of the LDA. From now on when referencing the standard output of the model we mean the two matrices. We might also references simple derivates or additionally computed objects.

The training phase results in a LDA model which summarizes and explains the original corpus by discovering the underlying thematic structures and assigning topics to each document. This trained model can in turn be used to explore and understand new corpora. The reader should bare in mind that the posterior inference and the topic model at large are statistical processes and incoherent topics may arise out of them.

1.5 Patent Data

In order to organize patents and allow stakeholders to search, access and review them easily, patent documents are organized by national and international patent offices. In order to cope with the amounts of patents reaching those organizations, patents are organized in hierarchical classification systems like the International Patent Classification (IPC) which divides patents into over thousands of categories over four levels with ever refining scope. The classification is based upon the used technology as well as other patent related characteristics. In order to make patent data more accessible, patents can be accessed through databases where all the available information is split across different tables.

The structure of patent documents is of interest to the visualization task at hand since those documents provide sections with different types of information. Generally the sections of a patent or patent application will contain at least the following or similar sections:

- patent-id: unique identification code indicating the location inside the classification system
- title: descriptive titel of the patent
- patent holders: list of the patent proprietors
- abstract: resume of the description
- description: detailed description of the patents contents

Aforementioned contents of patent documents like the description and abstract can be used to train a topic model. These informations are unstructured data. Other information like the year of publication or the classification of the patent could be used to extend a visualisation by displaying them to the user. These informations are structured data. Some advanced topic models can also be trained using information like the year of publication or even the patent holders name in addition to the unstructured data. Rosen-Zvi et al. (2004) detail the use of the authors name in their research on the author-topic model.

2 Literature

Over the course of the following chapter the reader gets an overview of the different visualization techniques that can be found in the literature as well as the tools designed to visualize and explore topic models. We explain the works of two research papers treating on visualization techniques and detail the design of three others regarding visualization tools. We also give an overview of metrics developed to enhance visualizations and evaluate topic models. We have a deeper look at five research papers regarding metrics. Each segment is concluded by a paragraph putting the techniques and knowledge presented in the paper in the context of our implementation.

2.1 Visualisation

Boyd-Graber et al. (2017) give an excellent introduction to topic models. This work contains chapters on how to evaluate, interpret and visualize models. The authors also present use cases where researchers will find it useful to adapt their models to the given application.

In regards to the visualization techniques the authors distinguish the two separate tasks of visualizing the topics themselves and the topic model in general. In order to implement the first task the authors name the word list. This intuitive technique consists of displaying a list of the topics most probable words. The authors cite a handful of works applying minor variations to this technique. The other presented visualization technique is the word cloud (figure 2.1). In its basic variation, the words of a given topic are placed one next to the other forming a cloud of words. The size and colour of the words can be used to map the probability of the words and add additional information to the graph. According to the authors word clouds come with the drawback of being poorly suited for visual search and tend to lead to false conclusions due to the lack of contextual information. One of the cited works addressing this issue is Smith et al. (2014) which we will discuss later in this chapter. When it comes to displaying models we want to tie the model back to the original corpus. In order to undertake this visualization the authors propose linking the topics with the documents they are most strongly represented in. They also propose including meta-information from the corpus to the visualization like showing the evolution of topics over time or displaying to users how topics compare to one and another using a termite visualization.

In the context of this thesis we utilise the authors work as an introduction to word clouds. We also use the references to other work in order to extend the scope of our literature overview.

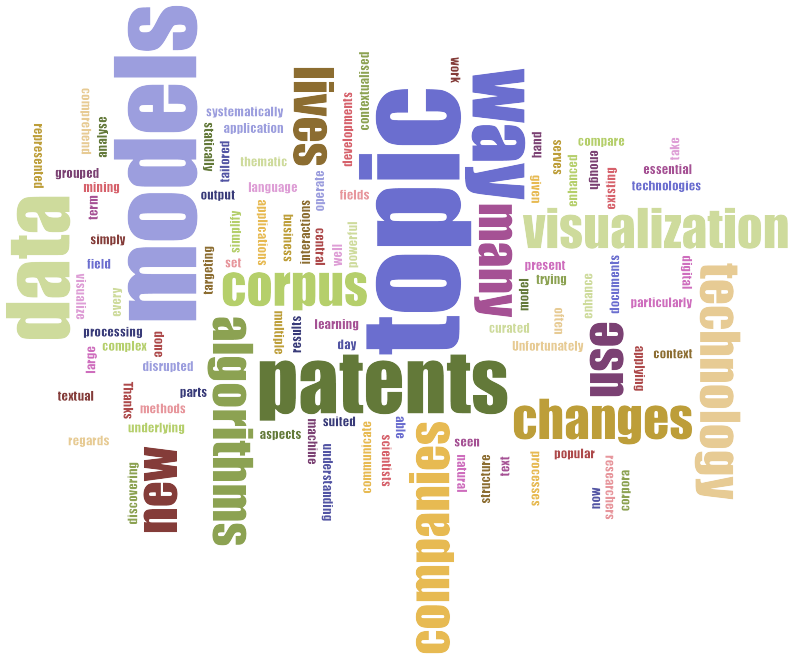


Figure 2.1: Word cloud generated from the motivation of this thesis (source: www.jasondavies.com/wordcloud/)

Smith et al. (2014) propose a term co-occurrence and a topic covariance measure which aim to allow users to judge the pertinence of a topic model. Both metrics can be computed without any changes to the topic model itself. The authors present a visualization technique integrating the aforementioned metrics. This work is meant as an improvement on word clouds.

The Pointwise Mutual Information (PMI) score is used to measure the association between two words of a document or corpus. This method has originated in other works and will be presented in greater detail in the next section 2.2. The authors propose representing topics as a force-directed network graphs. Words of a given topic become nodes of the network whose size represents their probability. In contrast to word clouds nodes of the same probability will then have the same size regardless of the words length. Edges between nodes become visible when the PMI score of the associated words surpasses a set threshold. In addition to the visual feedback the PMI weighs the edges resulting in clusters of words with high PMI scores inside the topics. The authors then propose to place the topics in boxes of a group-in-a-box layout. Using the topic covariance, similar topics can then be placed in proximity. The authors also include minor features like displaying information on hover or click events as well as an editing mode where users can correct bad topics.

Smith et al. (2014) is our most significant influence when implementing the visualization of

individual topics. We represent topics as network graphs and use the PMI score of word pairs and topics to enhance the visualization.

Chaney and Blei (2012) lay out their design for a topic model visualization tool. The design of the tool is based on the understanding that when a topic model is applied to a corpus it summarizes said corpus into the topics that are present in it. The topic model therefore allows the organization of the corpus and essentially becomes a taxonomy by theme of the corpus.

The tool consists of the two distinct visualizations: the topic and document views. Both views behave like types of pages on a website would. The first view consists of an ordered word list of the topics most probable words. In addition to the word list we can also view both the associated documents and similar topics to the one being currently viewed. The word list and documents are computed using the models standard output. In contrary, the related topics are selected using a pairwise topic dissimilarity score. The document view displays a list showing the topics the selected document consists of. Additionally the document itself and a list of documents having similar topical composition are displayed. The topic list and the similar document are computed using a topic models standard output. All views are linked between themselves allowing users to discover the corpus and topic model simultaneously.

We implement parts of the document view in the implementation of our tool. We will display documents associated to the explored topics and allow users to view portions of their content.

Gardner et al. (2010) present their design of a visualization tool called the Topic Browser. The tool incorporates a multitude of established techniques while the authors also propose new methods of visualization. The tool makes use of an assortment of metrics relating to the individual topics and documents. Additionally the authors incorporate meta-data related to the documents of the corpus.

In regards to the actual design of the tool the authors present three components with distinct functionalities. The first of these components is the side bar. It allows users to precisely filter by the metrics mentioned earlier and additional items. Both the metrics and items which users can filter by depend on whether topics or documents are currently being viewed. When users have selected the constraints they want to filter by they can browse the resulting topics or documents. Apart from an improved word cloud the topic view consist of a section showing users how the word of a topic are used in context inside the corpus. The topic view also allows the exploration of attributes present in the documents of a given topic and displays similar topics. On the other hand the document view consists of statistics computed for the viewed document. Users can also view a document by adding data from the topic model and overlaying it onto the document. The last feature of the Topic Browser are its plots. Users can plot the topics over document

attributes, essentially exploring the distribution of topics. Users can also plot the regression of two topic metrics.

We use the work of Gardner et al. (2010) as influence on incorporating information from the original corpus into the visualization tool.

In the works of Eisenstein et al. (2012) the authors detail their design of a visualization tool called TopicViz. The tool seeks to combine the capabilities of a search engine and the output of a topic model allowing users to explore document corpora more efficiently. This tool allows users to refine their search queries, understand topics and discover documents that are not in apparent connection to the search query.

When using the tool users can input a search query. The search engine's output is a document list with documents from the corpora that matched the users query. In the tool's graphing component the documents selected from the document list and the topics they are associated with are displayed in a force directed graph. By fixing either the documents or the topics in place the unfixed elements will be floating in between and around them. The unfixed elements position is determined by their attraction to the fixed elements. This way documents will be closer to the topics they consist of than the ones they do not. The authors present use cases.

We do not implement any part of this work into our visualization tool.

2.2 Metrics

Boyd-Graber et al. (2017) provide a historical overview of metrics quantifying topic and topic model quality. The metrics strong points and short comings are layed out by the authors. The works of Newman, Lau, et al. (2010), Mimno et al. (2011) and Lau et al. (2014) are cited by the authors in the context of automatically rating topics on their human interpretability. Those works will be examined later in this section. The authors also speak on the techniques implementing a feedback loop into visualizations to allow users to manually correct models.

Newman, Karimi, et al. (2009) introduce different metrics with the aim of scoring the usefulness of a topic. The score of a topic should be similar to the interpretability score given by human judges. In doing so the authors intent to automatically discover topics that may be statistically relevant but make no sense when interpreted by humans.

The metric which ranks closest to human scoring is the PMI score. In this work the authors use external data in order to eliminate statistical noise and unusual word statistics in the training corpus. The score of a word pair (w_i, w_j) is calculated by weighing the common probability of

both words with the product of the individual probabilities and logarithmizing the result. See the following formula (2.1):

$$PMI\left(w_i, w_j\right) = log \frac{p\left(w_i, w_j\right)}{p\left(w_i\right) p\left(w_j\right)} \tag{2.1}$$

Martin and Jurafsky (2009) describe the intuition behind PMI scoring as measuring how often two words appear together in the corpus and comparing it to how often they should appear together if they were independent. Following this intuition the denominator of formula 2.1 measures the actual appearances. The nominator measures the appearances under the assumption of independence. The granularity used when counting co-occurrences is variable. They can be counted inside individual documents or a word window. In order to calculate the PMI score of a topic the authors propose the median of the PMI scores of the topics most probable words. The PMI metric is compared to metrics based on the results of Google search queries using the topics most probable words. In addition to the topic evaluation metrics the authors also propose a count-based topic similarity score. When computed with topics having high PMI scores, this similarity metric shows promising results.

We implement the PMI metric presented in this work as a central component to our visualization tool. The metric is used to determine a topics usefulness and to give context to the words appearing alongside each other inside a topic visualization.

With Newman, Lau, et al. (2010) the authors extend the work of Newman, Karimi, et al. (2009). In this work the authors compare a multitude of new and more complex metrics measuring topic coherence and human interpretability. This work aims to find the metrics with the highest correlation to human scoring while allowing for a complete automation of the scoring task. The authors tested multiple metrics based on external data from Wikipedia articles, Google search results or the WordNet database. The tested scoring methods include the Path distance, the Milne-Witten score, Google title matches and many more. The overall results indicate that the PMI metric achieves better results than the other methods.

This work can be seen as validation of the choice of the PMI score as a metric for our visualization.

Mimno et al. (2011) explore the ways in which a topic might be flawed and propose a metric targeting those issues. This metric is then worked into the design of a topic model in order to minimize the creation of bad topics at the source.

The authors consider a topic to be flawed when one or more of the following issues is present: chained or intruded words are present, the topic is random or unbalanced. Chained words are words from distinct themes still present in the same topic by being associated to one or more

common words. Intruded words are similar in that they stem from unrelated themes. In contrast intruded words have no relation, they are not chained. An unbalanced topic consists of related words with varying scopes: specific words of a theme are associated to general terms. The authors found that those issues where detectable using a metric based on word co-occurrence inside the corpus. The Generalized Pólya Urn Model is proposed to incorporate said metric in order to enhance the quality of topics in the generating process. The model essentially limits the creation of topics where words that are the most probable rarely co-occur with other words of high likelihood.

Lau et al. (2014) explore ways in which to automate the evaluation of the intrinsic quality of individual topics and topic models. The metrics are tested on the word intrusion and observed coherence tasks which have both been put forward to compute topic interpretability. For both topic and model levels the authors propose ways to automate the tasks and compare them against human scoring with promising results.

3 Tool Design

In the following chapter we go over the tool implemented alongside this thesis. We start by defining the constraints and requirements dictating the overall design of the tool. We then proceed by explaining the design choices taken during the implementation process. This section details the inner workings of the tool and the challenges in implementing it. This chapter is concluded by an explanation of the tools components as well as their function and usage.

3.1 Requirements

The development phase follows the spiral model. This way we can build the simplest working visualization tool and iteratively incorporate new features. Even though this is a small software project proceeding this way anchors the development process and allows us to have a working implementation at any stage. Additionally when errors occur they can be easily detected and rolling back does not result in a broken tool.

We have set general requirements that our tool has to fulfil. Firstly, the tool has to be written in the Python programming language. This requirement guarantees that the tool is serviceable and understandable by most people. We also require the tool to allow and be able to handle user input. The primary user input are the parameters that serve as constraints for the different visualizations. Additionally we want to implement visualizations for the three different levels of a topic model. Those levels are the individual topics, the model in general and the documents used to train the model. Finally, we expect the tool to be practical i.e. it should be reactive, intuitive and extendable in its features.

In addition to the general requirements to the tool we have set requirements to the three levels of visualizations we seek to implement. Concerning the view displaying individual topics we define Smith et al. (2014) as our major influence. Like it is described by the authors, topics are visualized as network graphs where words take on the form of nodes. The edges between the word nodes convey the PMI score of the word pair. A PMI score is also generated for each displayed topic. The user can utilise this view to get an understanding of what the topics consist of. This view should also reveal a topics quality. In regards to visualizing the model in general we require that the tool features a view combining information gained trough the topic

model with the original corpus. This view allows the user to gain insights into the distribution of topics inside the corpus. The user can also decide which topics to investigate. When it comes to the visualization of documents, the associated information and external sources we do not take influence from the literature. We let the possibilities of the programming libraries we choose guide the design of this view. With this view we aim to display interesting information originating both from the source used to train the topic model and additional sources.

3.2 Implementation

The first choice in regards to the fundamental design of the tool is to make it take the form of a dashboard. As we stated in the requirements, the tool is programmed in Python. By implementing both fundamental design facets we ensure that the tool is easily deployable on a local machines browser as well as on a server making it accessible to the entire Chair in Economic Policy. We implement the dashboard using the dash library which combines HTML, CSS and JavaScript capabilities. We use the plotly graphing library to implement the visualizations of our tool. Both libraries are developed by Plotly and therefore integrate seamlessly.

In the next section we detail the design and structure of the tool itself.

The dashboard is build as a Python coded HTML structure. Similarly to programming normal websites we organize the tool into containers and elements. The elements are grouped by their function or the visualization they relate to. The containers enclose the different elements and are arranged according the way the dashboard should look like. Both the elements and their containers have attributes that can be set statically or through user interaction. Certain elements of the dash library are specifically designed to allow for user interaction like radio buttons or range sliders. In our implementation we use those elements to manage the parameters a given user wants to set for the visualizations or other features. In order to connect the given parameters to one or more visualizations we use a feature of the dash library: callback functions. Callback functions take at least one parameter like the value of a text field and proceed to perform operations on their parameters and other structures of the code. After having performed the operations a callback function returns updated attributes of an element. The tools structure of containers and elements can be found from line 60 to 207 of A.1. All the tools callback functions are programmed from line 209 to 492 of the same file. Helper functions used for minor tasks can be found in A.2.

Our implementation consists of elements which only function as inputs for the visualizations i.e. as the parameters of the graphs and other elements. The visualizations rely on their parameters and separately computed data to run their callback function. In our implementation the attributes of the network graph might be the output of a callback function. In this case the attributes might

be the new positions of nodes and edges inside the graph. Coupling the structure of the app with the callback functions allows to precisely manage user interactions.

The tool requires both matrices that form the output of a topic model (section 1.4). In addition, the tool requires a dictionary mapping words to their id in order to revert the ids we find in the standard output back into text. We also make use of the dataframe the topic model was originally trained with. We make use of this data in the visualization. The tool also depends on two objects containing the word and pair counts of the original corpus. Both objects can be computed with A.3 provided with the thesis. We explain the content of those objects in the next paragraph. The code importing the necessary files and objects can be found from line 20 to 52 of A.1.

During the implementation of this tool, amongst the usual complications, two notable challenges arose which greatly influenced the tools outcome and its design. The first challenge in implementing this tool is the computation of the PMI score for all the word pairs that are displayed at a given time. Indeed computing the word and pair counts as well the PMI scores at run time results in bad usability due to longer loading times in between visualizations. In order to decrease the delays in between visualizing new topics or computing new settings we pre-compute the PMI scores for the most probable words of all topics. The word and word pair counts of the corpus are also computed pre-emptively. We consider two words to co-occur when they appear in the same document. The script A.3 which is responsible for those computations also stores the resulting objects locally. When displaying the scores we verify if a given score has already been calculated. If it has not we compute it at runtime, otherwise we retrieve it from the saved object. We use formula 2.1 for both the pre-emptive and runtime PMI calculations. The second challenge in implementing this tool is the lack of data contained in the dataframe originally used to trained the topic model. When designing and implementing a topic model only some data is required. In order to augment the information available to the user we establish a connection to an external data source. We use the patent-ids stored in the original dataframe object to query the Chair of Economic Policy's patent database at runtime. The data received as a result of the query is then used to enhance the tool.

The realization of the design choices described in this section lead to a fast and reliable tool which is suited for exploring topic models and the corpora their are trained with. The underlying HTML structure and the concept of callback functions make the tool extremely modular. The user can easily extend the capabilities of the tool by implementing new elements or enhancing and modifying the current ones.

3.3 Tool Overview

In the following section we describe the components and views featured in our tool and detail their use. This overview is divided into four segments. In the first segment we describe the components which allow users to input the parameters used for the visualizations. In the last three segments we describe the different visualizations featured in the tool. All the screenshots displayed in this section are the result of the same visualization parameters.

The parameter box (figure 3.1) is where parameters used for the visualization are set by the user. The *Topic Multi-Select Dropdown* serves to select the topics to graph in the topic view (figure 3.2). The user can select one or more topics trough the menus search functionality. The amount of words and the range of scores shown can be set using the *Word Slider* and *PMI Slider*. By carefully choosing the PMI score range to display the user can discover inconsistencies between the words of a topic indicating a faulty topic. Both Sliders feature indicators displaying their setting in order to facilitate their usage. The *Nb of Words for Topic PMI score* button sets the amount of most probable words that are used to calculate the topics median PMI score. This feature could be used to discover the words that best describe the topic while still being coherent and representative of the topic in the corpus. The *Nb of Documents* button sets the number of documents we want to display for the selected topics. Those documents are displayed in the table view (figure 3.3). Contrary to the other elements of the parameter box this button triggers a callback to the table view as soon as its value is changed. When the *Display Visualization* button is triggered the callback functions of the graph and table view are called with the current settings of the parameter box. This means that the user can set the parameters without being interrupted and proceed to the visualization when he chooses to.

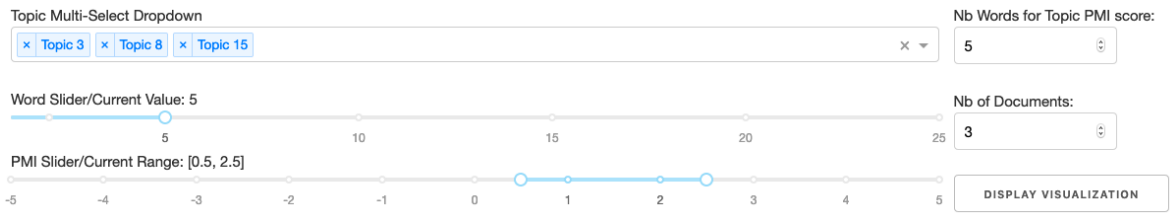


Figure 3.1: The parameter box allowing the user to set the parameters of the visualization. (source: screenshot of the tool)

The topic view (figure 3.2) is where the user can see the visualizations of the topics and settings selected inside the parameter box. The callback function computing this view is build as an iteration over the topics selected in the topics dropdown menu. We build a network graph for each topic with the selected number of words. We then compute the PMI score for all possible word pairs. When a word pairs PMI score is stored in the saved dataframes this one is used. Otherwise the PMI score is computed at runtime. When the PMI score is retrieved or computed it is compared to the selected range in the parameters. If it is within the range the edge

relating to the scores word pair is drawn and the score is annotated. Otherwise the word pair remains unconnected. The nodes of a topic are then positioned using the Fruchterman-Reingold force-directed algorithm. After this the topics median PMI score is computed using the n most probable words set in the parameters. The topics median PMI score are displayed on the top left corner of a topic. The plotly graphing library enables, amongst others, functionalities like zooming into a topic or saving the visualization as a picture. By using this topic visualization the tools user can get an understanding of the topic models most basic output.

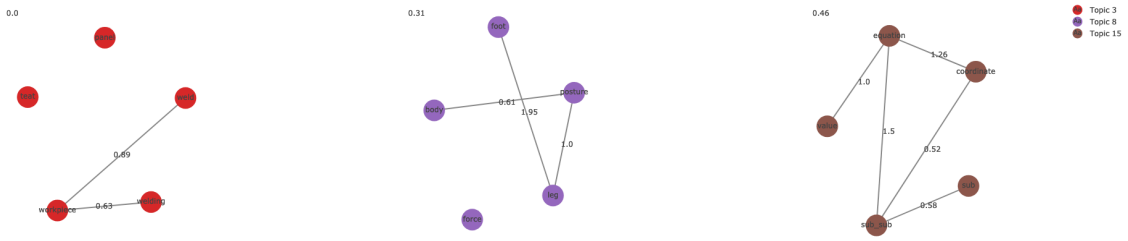
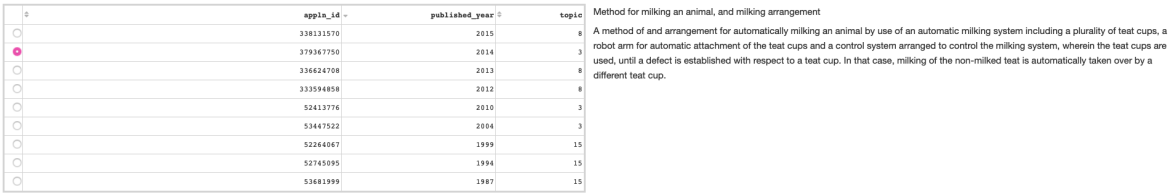


Figure 3.2: The network graphs displaying the selected topics. (source: screenshot of the tool)

We use a traditional table view (figure 3.3) to show the user the top documents associated with the selected topics. When the display button is triggered we select the documents that are most representative of the topics that are selected in the dropdown menu. To select those documents we use the document and topics matrix from the models standard output. The documents are then matched against the original dataframe and we display their applicaton id, year of publication and the topic they are associated with in the form of a table. The table allows sorting the columns and selecting one row at a time. When the user selects the row of a document he is interested in, two SQL queries to the Chair in Economic Policy's patent database are triggered. The titel and abstract of the document are the answers to these queries and are displayed as Markdown to the right of the table. By using this visualization the user can put the selected selected and the displayed words into context. The table view allows for the user to situate the topic in time and inside the patent classification system. The Markdown allows the user to identify the patents content further deepening the insights into the topics.



	appln_id	published_year	topic
○	338131570	2015	8
●	379367750	2014	3
○	336624708	2013	8
○	333594858	2012	8
○	52413776	2010	3
○	53447522	2004	3
○	52264067	1999	15
○	52745095	1994	15
○	53681999	1987	15

Method for milking an animal, and milking arrangement

A method of and arrangement for automatically milking an animal by use of an automatic milking system including a plurality of teat cups, a robot arm for automatic attachment of the teat cups and a control system arranged to control the milking system, wherein the teat cups are used, until a defect is established with respect to a teat cup. In that case, milking of the non-milked teat is automatically taken over by a different teat cup.

Figure 3.3: The table displaying the top documents associated to the selected topics. The text displaying the title and abstract of the selected patent. (source: screenshot of the tool)

We choose to overlay the topic models output onto the corpus using a treemap (figure 3.4). This visualization does not rely on any user interaction and is not linked to the parameters box. This display is computed one time at runtime when the tool is launched. We use the document and topics matrix to compute the topical composure of the corpus. The percentages of a given topic

are summed over all documents. In the treemap each square represents a topics proportion inside the corpus. By using this visualization the user can get an overview of the corpus and start to look for interesting topics.

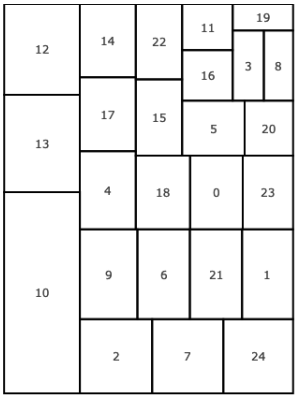


Figure 3.4: The Treemap showing the distribution of topics accross the corpus. (source: screenshot of the tool)

4 Conclusion

In the following chapter the reader will be presented with a summary of this thesis work. This summary will reference the research questions guiding this work (section 1.2). We also give an outlook on the possibilities to enhance the tool.

4.1 Summary

The visualization of topic models and especially those trained with patent data remains a poorly investigated field. Over the course of this thesis we took a look at the current developments of visualization techniques and tools used to display topic models. We examined techniques targeting the different levels of topic models: exploring word lists, clouds and termite visualizations. We also presented metrics like the PMI score used in the evaluation task of topic models and as enhancement to visualizations. Metrics used in the measurement of topic resemblance where also cited. Further investigating the techniques and metrics applied in the context of visualization tools and the tools design allowed us to get an understanding of the holistic visualization of topic models.

We also defined the optimal visualization of topic models trained on patent data. This visualization should consist of a holistic view: giving an overview of the entire corpus in relation to the discovered topics, displaying one or multiple topics using an enhanced word-cloud-like visualization and finally bridging the topic model and the training corpus. The visualization should also make use of appropriate metrics in order to evaluate the computed topics while enhancing the visualizations. In order to accommodate for topic models trained on patents, the visualization should encompass the structured data in addition to the unstructured data which is used for training the model. Therefore the tool should be able to link to databases containing the patents and additional informations. This connection should be used to add interesting information to the visualization like a patents abstract or the classification hierarchy of the patents used in training the model.

As part of this thesis we tried to implement the best visualization tool we could while still fulfilling the aforementioned requirements. As our view displaying individual topics we chose to implement our interpretation of Smith et al. (2014) work. When displayed in our tool, topics

take on the form of network graphs enhanced by the use of the PMI scoring method. Additionally we display patent information obtained through the corpus and the Chair in Economic Policy's patent database. The corpus is also displayed through the discovered topics using a treemap visualization conveying a the topics importance in the explored corpus. The created tool is by no means perfect but it is lightweight, modifiable and intuitive in it's use. The visualization tool also presents a good platform from which to implement new visualizations.

4.2 Outlook

Innovations concerning the technical processes behind topic modelling will continue to emerge from the scientific community. In order to convey the insights of those novel topic models the technical innovations will have to find their way into the visualization tools. With the tool created alongside this thesis, we have laid the foundation necessary to visualize those innovations. Due to its modular nature the tool is perfectly suited the handle tomorrows topic model.

Even without the newest publications there are multiple ways to increase the tools usability and versatility. One could envision different ways to visualize the individual topics. In addition to the network graphs a termite visualization could be implemented. The current representation could also be combined with a treemap in order to add information to the visualization through the squares size, colour and position. Users could also implement advanced evaluation scores to enhance the interpretability of words inside a topics or better score the topics quality. The topic view could also display similar topics to the ones being displayed. We could also envision displaying additional information relating to the corpus used to training the topic model. In this regard displaying statistics relating to the corpus like the topical composition of the documents comes to mind. The treemap view could also be improved by using colours to display additional information. In this context the colours could be used to convey the topics coherence.

Between the innovations coming out of the scientific community and the improvements already imaginable this tool seems like an endless project. In the end the only limit is the imagination of the programmer improving the tool.

A Appendix

This section serves as reference to the Python files submitted alongside the written thesis.

A.1 visualization.py

Python file containing the tools structure and major functions: visualisation.py

A.2 utility.py

Python file containing helper functions performing minor tasks for the creation of the visualization: utility.py

A.3 pmi.py

Python file containing the script which computes and stores the word and pair count as well as the PMI scores of all word pairs for each topic: pmi.py

References

Blei, D. M.; Ng, A. Y.; Jordan, M. I. (2003): Latent dirichlet allocation. In: *Journal of machine Learning research* 3(Jan), pp. 993–1022.

Boyd-Graber, J.; Hu, Y.; Mimno, D., et al. (2017): Applications of topic models. In: *Foundations and Trends® in Information Retrieval* 11(2-3), pp. 143–296.

Chaney, A. J.-B.; Blei, D. M. (2012): Visualizing topic models. In: *Sixth international AAAI conference on weblogs and social media*.

Eisenstein, J.; Chau, D. H.; Kittur, A.; Xing, E. (2012): TopicViz: interactive topic exploration in document collections. In: *CHI'12 Extended Abstracts on Human Factors in Computing Systems*. ACM, pp. 2177–2182.

Gardner, M. J.; Lutes, J.; Lund, J.; Hansen, J.; Walker, D.; Ringger, E.; Seppi, K. (2010): The topic browser: An interactive tool for browsing topic models. In: *NIPS Workshop on Challenges of Data Visualization*. Vol. 2. Whistler Canada.

Hall, B. H. (2004): Exploring the patent explosion. In: *The Journal of Technology Transfer* 30(1-2), pp. 35–48.

Lau, J. H.; Newman, D.; Baldwin, T. (2014): Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 530–539.

Martin, J. H.; Jurafsky, D. (2009): *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson/Prentice Hall Upper Saddle River.

Mimno, D.; Wallach, H. M.; Talley, E.; Leenders, M.; McCallum, A. (2011): Optimizing semantic coherence in topic models. In: *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, pp. 262–272.

Newman, D.; Karimi, S.; Cavedon, L. (2009): External evaluation of topic models. In: *in Australasian Doc. Comp. Symp., 2009*. Citeseer.

Newman, D.; Lau, J. H.; Grieser, K.; Baldwin, T. (2010): Automatic evaluation of topic coherence. In: *Human Language Technologies: The 2010 Annual Conference of the North*

American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, pp. 100–108.

Rosen-Zvi, M.; Griffiths, T.; Steyvers, M.; Smyth, P. (2004): The author-topic model for authors and documents. In: *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, pp. 487–494.

Smith, A.; Chuang, J.; Hu, Y.; Boyd-Graber, J.; Findlater, L. (2014): Concurrent visualization of relationships between words and topics in topic models. In: *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pp. 79–82.