University of Zurich[UZH]

Topic Extraction and Visualisation of Digitalisation Related Research from ZORA

Thesis January 31, 2018

Lukas Vollenweider
of Zurich, ZH, Switzerland

Student-ID: 13-751-888
lukas.vollenweider@uzh.ch

Advisor: **Bibek Paudel**

Prof. Abraham Bernstein, PhD
Institut für Informatik
Universität Zürich
http://www.ifi.uzh.ch/ddis

Acknowledgements

I would like to thank my supervisor, Bibek Paudel, for helping me a lot by sharing his knowledge about topic extraction and data mining in general as well as always giving valuable feedback on how things could be improved.

I would also like to thank to Prof. Sara Fabrikant, Prof. Abraham Bernstein and Sarah Lechmann for giving valuable feedback on the feature list of the web application.

Finally, I would like to thank to Prof. Abraham Bernstein for making it possible to gain deeper knowledge in the data mining process by letting me work on this project as my bachelor thesis.

Zusammenfassung

Durch die stetig wachsende Zunahme an verfügbaren Dokumenten im Internet werden Methoden benötigt, die helfen den Kontext der Daten zu ermitteln, ohne alles zu lesen. Solche Methoden, Topic Models, existieren bereits, funktionieren aber meistens nur für grosse Dokumente. Diese Arbeit analysiert die momentan benutzten Topic Models, präsentiert aber auch zwei neue, kontextsensitive Möglichkeiten anhand eines Datensatzes der auf Zusammenfassungen basiert. Anschliessend werden die besten Resultate visuell aufbereitet um die Interpretierbarkeit zu verbessern.

Abstract

Due to the fast increasement of available documents in the Internet, methods are needed which are able to present the content of the data, without the need to read them. This methods already exists, called topic models, but tend to work only for large documents. This work analyses current state-of-the-art topic models as well as presenting some own, context-sensitive approaches on a restricted data set built from abstracts. Then, the best results are visualised to improve the interpretability of the data.

Table of Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Description of Work . . . 2
- 1.3 Outline . . . 2

2 Preliminaries and Related Work **5**
- 2.1 Topic Modelling . . . 5
  - 2.1.1 LDA . . . 7
  - 2.1.2 NMF . . . 9
- 2.2 Topic Modelling on Abstracts . . . 10
- 2.3 Recent Advances in Natural Language Processing . . . 11
- 2.4 Evaluation of Topic Models . . . 12
- 2.5 Visualisation of Topic Models . . . 13

3 Data Analysis **15**
- 3.1 Data Quality . . . 15
- 3.2 Preprocessing . . . 19
  - 3.2.1 Removing stop words . . . 19
  - 3.2.2 Removing noise . . . 21

4 Topic Model Evaluation **23**
- 4.1 Evaluation Techniques . . . 23
- 4.2 LDA . . . 24
- 4.3 NMF . . . 27
- 4.4 Doc2Vec + k-Means . . . 30
- 4.5 CNMF . . . 34
- 4.6 Discussion . . . 40

5 Visualisation **43**
- 5.1 Reducing Dimensions . . . 43
- 5.2 Topic Labelling . . . 45
- 5.3 Web Application . . . 46

6 Future Work **55**

7 Conclusions **57**

A Appendix **63**
A.1 Example of a JSON export from ZORA 63

B Appendix **65**
B.1 CD Content 65
B.2 Installation Guide 65

1

Introduction

We live in a time where the amount of data available for people to read increases with every second. According to "Internet Live Stats"[1], nearly 8000 Tweets and over 2.5 million emails are sent in just one second. According to Forbes[2], in the two years from 2013 up to 2015, more data has been created than in the entire previous history of the human race. They expect that by the year 2020, about 1.7 megabytes of new information will be created every second for every human being on the earth. Since the world wide web became a substantial element of our daily lives, we're being flooded by data. However, not all of the data being created is of use. But how can we decide, which documents, which emails, which tweets are worth being read, which article worth being bought with such an overdose of data available on the Internet? The need for methods, which can automatically capture the main topics of a set of documents arises more than ever.

1.1 Motivation

The UZH Digital Society Initiative (DSI)[3] is a network of currently about 150 researchers located around the University of Zurich (UZH). They provide an academic platform for promoting critical, interdisciplinary reflection and innovation on all aspects of the digitalisation of society and the sciences. Since they host many different people from different fields of interest, an overview about what they have written about and what topics currently are not covered by them, is not available. These 150 people have published more than 7000 documents on ZORA (Zurich Open Repository and Archive), a collection of documents which is too large to analyse by hand.

Topic Models. Topic models are currently used for classifying documents into certain clusters. They then provide a list of keywords, which should make it easier for people to understand the content of the analysed documents, without reading them. The aim of a topic model is to provide answers about big-picture questions of a large

[1] http://www.internetlivestats.com/one-second/, accessed on 23 Jan, 2018

[2] https://www.forbes.com/sites/bernardmarr/2015/09/30/big-data-20-mind-boggling-facts-everyone-must-read/#7f6ed2a317b1, accessed on 23 Jan, 2018

[3] http://www.dsi.uzh.ch/en.html, accessed on 24 Jan, 2018

document or collection of documents in a cheap, quick manner without human intervention [Boyd-Graber et al., 2017]. This sometimes works great if the extracted keywords are good enough. However, if they are not, it may be more more useful to analyse the underlying representation a document gets through the topic modelling process (e.g., how much a document belongs to each available topic) through visualisation, as [Bruggmann et al., 2013] did with wikipedia articles.

1.2 Description of Work

This work aims to provide an overview of what the researchers at DSI wrote about by utilising topic models. This is achieved by providing a topic extraction and visualisation framework, which will be the main output of this work. For the topic extraction part, the data set, restricted to the abstracts of the available papers of researches, which are part of the DSI, has to be thoroughly analysed. Based on this analysis, the data has to be preprocessed to remove as much noise as possible. Otherwise, the topic models may fail to produce valuable, interpretable outputs. As there are many available topic model algorithms, this work has to focus itself on only some of the available models. This was done on a first evaluation step, which is not further described here, where some models, like the ones mentioned in section were tested but deemed to be not good enough to use it for the topic model framework. The basic two state-of-the-art topic models, NMF and LDA, were two topic models which worked rather well on the data set and therefore, are analysed further in this work. With the hope of improve on NMF and LDA, two additional topic models are analysed, Doc2Vec + k-Means and CNMF, a combination of NMF and a convolutional neural network (more in Section 4.5), which were specifically developed for this work. For visualisation, a javascript web application was created with the goal of providing an interactive environment for people to discover the extracted informations by the topic model. The visualisation part is influenced by the work of [Bruggmann et al., 2013, Bruggmann, 2017]. The web application can be accessed at: *https://rebrand.ly/dsi_topic_extractor*.

1.3 Outline

First, Chapter 2 will go through the current state-of-the-art topic models as well as introducing some notation needed to understand these models. Since the data set is restricted to the abstracts, section 2.2 will present some work on abstract-based topic modelling. While none of the presented methods are considered for the topic model analysis, since they did not perform well enough on the data set, they still provide valuable input on the problem of using abstracts as a data source in general. Since one challenge of topic modelling is to evaluate the result, section 2.4 presents a method which can be used to evaluate the coherence of keywords from a topic. Chapter 2 concludes by presenting some methods and examples about possible visualisation techniques, as well as presenting a potential

Chapter 3 will give an overview about the data used in this work. First, the language distribution as well as potential natural clusters are analysed. The Chapter concludes by explaining the techniques used to preprocess the data for the topic models.

Chapter 4 and 5 are the main parts of this work. Chapter 4 evaluates four topic models on the data set: LDA, NMF, Doc2Vec + k-Means and CNMF. For each of the algorithms, the extracted keywords as well as the institutes which contributed the most per topic are analysed. The Chapter concludes with a discussion based on this analysis, as well as the PMI score explained in section 2.4. Chapter 5 presents two techniques on how the document representations resulted from the topic modelling algorithm can be mapped to a two-dimensional coordinates. Then, an algorithm is presented which is used to improve the topic description, since the extracted keywords might not be that easy to interpret. The Chapter concludes by guiding through the web application. This work is concluded with Chapter 6, where potential improvements are proposed and Chapter 7, where a conclusion about the work done is drawn.

2

Preliminaries and Related Work

Analysing the content of documents is a very broad area. There are many different use cases where one would like to know what a document contains without reading it, ranging from gathering market intelligence from news sources on the web to automatic spam filtering of e-mail contents [Weiss et al., 2004]. In Section 2.1 we give an in-depth look at two of the currently most used algorithms for topic modelling: LDA (Section 2.1.1) and NMF (Section 2.1.2). In Section 2.2, the potential problems on performing topic modelling on abstracts are highlighted. In Section 2.3, some of the most recent advances in NLP are highlighted. Then, in Section 2.4, we discuss ways to evaluate topic models and finally, in Section 2.5, we present some of the most used technologies for visualising the results from topic modelling.

2.1 Topic Modelling

Of particular interest for this work are algorithms which categorises texts in a unsupervised manner. Topic models are able to group a number of documents into clusters and extract keywords as representations of these clusters. The following terms which are defined here are used throughout this work:

- A word w is the basic unit considered.
- A vocabulary V is a set of unique words. w_i is the i-th word in V.
- A document is a sequence of words denoted by $d = (w_1, w_2, ..., w_n)$, where n is the number of words in the document.
- A corpus is a set of documents denoted by $C = \{d_1, d_2, ..., d_m\}$, where m is the number of documents in the corpus.
- A document matrix D represents the words of the documents with numeric representations. Each of the rows represents a document, each column a vocabulary entry and each cell is a value which represents the impact the word has on the document. This may be weights (e.g., TF-IDF factors, more on that below) or the number of times a word appears in all documents. The dimension of the matrix D is $m \times |V|$ and each column represents a word from V. This is a bag of words model since the order of the words are dropped.

Input Restrictions. Given a document matrix $D \in \mathbb{R}^{m \times |V|}$, the input is expected to have the following format, after some preprocessing is applied to V, such that all elements of V consist of only letters (Section 3.2):

For a probabilistic topic model (more on that in Section 2.1.1), the document representation d is restricted as following:

$$\forall d_i \in D, d_i = \{t_1, t_2, ..., t_k\}, \text{where } k = |V|,$$

t is the number of occurrences in each document $d \in D$ of the word identified by position i.

For a non-probabilistic topic model (more on that in section 2.1.2), the document representation d is restricted as following:

$$\forall d_i \in D, d_i = \{t_1, t_2, ..., t_k\}, \text{where } k = |V|,$$

t is a weight of the word identified by position i.

One important characteristic of the input D is, that it tends to be very sparse due to the very nature of the document representation: since every document is represented by a bag of words with size $|V|$, a lot of the entries of $d_i \in D$ will be 0.

Output. For a number of topics T, topic models output two different kind of information which are of interest: Topic models return a number l of words $w \in V$ for each topic t. This words are sorted by the importance they have in the detected cluster But topic models also generate a document-feature matrix $W \in \mathbb{R}^{m \times T}$, where

$$a_{ij} \in W$$

where m is the number of documents in the corpus and T is the number of topics. For probabilistic models, a is the percentage, with which document i belongs to topic j. For non-probabilistic models, a is a weight, document i has for topic j.

TF-IDF. One way to create a weight for a term $t \in V$ is to apply TF-IDF. TF-IDF stands for term frequency - inverse document frequency and is a weight which qualifies the influence the term frequency has. It consists of two components:

- Term Frequency: The term frequency measures how often a term occurs in a document. A good practise is to normalise the term frequency, since a term probably occurs more often in a long document than in a short one. The term frequency is calculated the following way:

$$\frac{\text{Number of times term t occurs in a document d}}{\text{total number of terms in d}}$$

- Inverse Document Frequency: The inverse document frequency punishes terms which occur in a lot of documents. The intuition behind it is that terms which occur in almost every document are less important to characterise a document

then terms which only occur in some documents. The inverse document frequency is calculated the following way:

$$log_e \times \frac{\text{Total number of documents}}{\text{Number of documents in which term t occurs}}$$

The weight resulting from Term Frequency * Inverse Document Frequency is used to measure the importance a word has in a given set of documents [TF-IDF, 2017].

2.1.1 LDA

The idea behind LDA is, that each document can be represented as a random mixture over latent topics, where each topic is characterised by a distribution over words. It is assumed that each document $d \in C$ can be generated from the following process:

1. Choose N $\sim$ Poisson(ξ)
2. Choose $\theta \sim$ Dir(α)
3. For each of the n words w_n:
   a) Choose a topic $z_n \sim$ Multinomial(θ)
   b) Choose a word w_n from $p(w_n \mid z_n, \beta)$

[Blei et al., 2003b]. This means, that a document d is seen as a random sequence of words w. Since each of the documents $d_m \in C$ can have multiple topics, a Dirichlet distribution is used. θ is the topic distribution for a specific document d, with hyper-parameter α as the Dirichlet prior. Therefore, after step 2, there is now a document which belongs to multiple topics with certain percentages. For each slot this document has (N in total), a word is selected the following way: a topic is chosen from the documents topic distribution, then a word is chosen which belongs to that topic from another Dirichlet distribution with hyper-parameter β as the Dirichlet prior. Since these two Dirichlet distributions are not really visible from the data, the algorithm is called latent Dirichlet Allocation.

Figure 2.1 shows the topic model. There are N words w, appearing in M documents. For each word w, a topic z is assigned. θ represents the topic distribution for a document $d \in M$. The only observable feature of the model is the word w, all of the other elements are inferred by LDA. The model also has two hyper-parameters, α and β. α is the prior to the Dirichlet distribution over documents, whereas β is the prior to the Dirichlet distribution over words [Blei et al., 2003b].

The algorithm. First, the algorithm iterates through all documents $d \in C$ and assigns to each $w_n \in d$ a random topic. Then, in an iterative process, each of the topic assignments is improved in the following way: For each word $w_n \in d$ and each document $d \in C$, the algorithm computes

$$p(t \mid d) \text{ and } p(w_n \mid t)$$

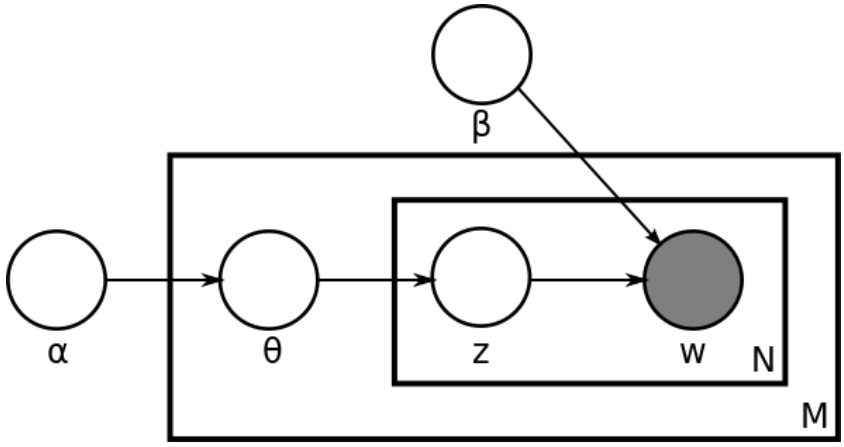


Figure 2.1: The LDA topic model, represented as a plate notation. From [Commons, 2008]

The first term is the probability on how much d belongs to t and the second term is the probability on how much w_n belongs to t. Under the assumption that all topics assignments for all words $w \in V \setminus \{w_n\}$ are correct, the topic of w_n is reassigned the following way:

$$t_{w_n} = p(t \mid w_n, d) = \frac{\#\text{ of } w_n \text{ in } t + \beta_{w_n}}{\text{number of words in } t + \beta} * (\#\text{ words in } d \text{ that belong to } t + \alpha)$$

After a few hundred iterations, the algorithm may be reaching a roughly steady state and the resulting document-topic matrix W, where each entry of a row represents a percentage on how much a document d belongs to a topic t and where each entry of the column from the topic-word matrix H represents a percentage on how much a word w belongs to a topic t [Underwood, 2014, Chen, 2011].

Characteristics. Since the documents are seen as a generative probabilistic process of choosing models from a Dirichlet distribution, LDA is seen as a probabilistic topic model [Blei, 2012]. This has the consequence, that the topic model relies on word counts to calculate certain probabilities, as can be seen in the algorithm part above. The consequence of this is that TF-IDF cannot be used as word representation in the bag of words model and therefore, LDA cannot use the word importance factor, which tends to be a valuable metric. However, since the entries of the resulting matrices W and H are also probabilities, this makes the matrices very easy to interpret, as shown above.

As the words are initialised with random topic probabilities, every run of LDA will produce different results and therefore, LDA only guarantees to converge at a local minima. Research shows that LDA is in general relatively unstable which may result in the need of running multiple runs of LDA to get satisfactory results [Koltcov et al., 2014].

2.1.2 NMF

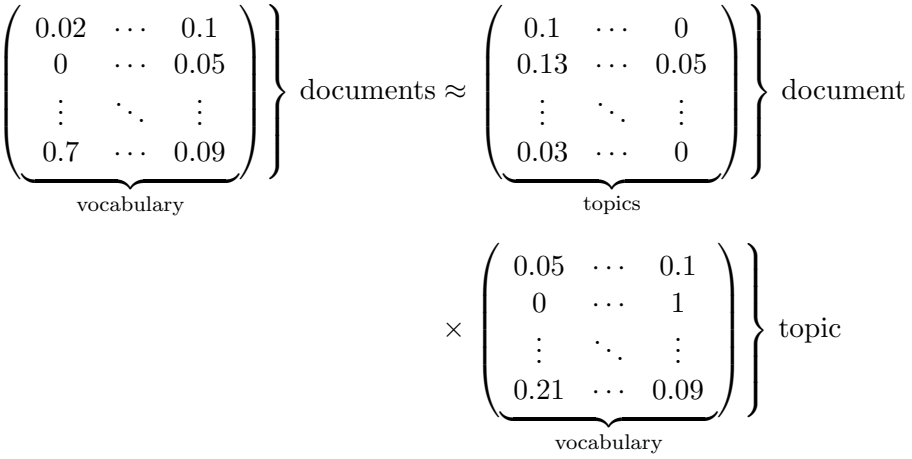


Figure 2.2: Concept of NMF for Topic Modelling

While NMF is not restricted to topic modelling, NMF for example does a good job at classifying images [Lee and Sebastian Seung, 1999], this work will only consider NMF as a topic model.

Other than LDA, NMF is a non-probabilistic approach for grouping documents into topics. The goal of NMF is to factorise the document matrix D into two matrices W and H such that

$$\begin{aligned} D \approx WH, \text{ where } D &\in \mathbb{R}^{|D| \times |V|}, \\ W &\in \mathbb{R}^{|D| \times T}, \\ H &\in \mathbb{R}^{T \times |V|} \end{aligned}$$

where D is a document-vocabulary matrix respecting the constraints defined in section 2.1, V is the vocabulary and T is the number of topics. Therefore, W is a document-feature matrix and H is a feature-word matrix. This factorisation is shown in 2.2. Since NMF is non-probabilistic, the entries of the document-vocabulary vectors $d_i \in D$ will often be weights calculated by TF-IDF.

The intuition behind NMF is that there are a T intrinsic topics in a set of documents and therefore, it should be possible to split D into two matrices which are connected by these topics without loosing any information.

The algorithm. At first, the two matrices W and H are initialised. They can either be randomly initialised or other, more sophisticated initialisation strategies can be used. Afterwards, the distance between D and WH is minimised by optimising a distance function over a few hundred iterations. One of the most used distance function is the frobenius norm:

$$F(W, H) = \|D - WH\|_F^2$$

As soon as the algorithm converges, the iterations can be stopped.

Characteristics. Both of the factorised matrices W and H are constrained to be non-negative (hence the name):

$$\forall a_{ij} \in W : a_{ij} \geq 0$$
$$\forall b_{ij} \in H : b_{ij} \geq 0$$

NMF does also not guarantee to converge to a global minima, therefore, if the two matrices W and H are initialised by random factors, it may be that they will converge to a local minima, which makes the algorithm unstable. A lot of thought has gone into how the stability of NMF can be increased. One way to improve the model is to use a different variant, called NNDSVD, which is suitable for sparse input matrices. This initialisation does not contain any randomisation in its basic form and therefore, converges to the same solution every time. [Boutsidis and Gallopoulos, 2008].

Since NMF is not a probabilistic model, the resulting row vectors from W do not sum up to 1. This makes the result of NMF hard to interpret. While higher values for a row in W mean that a document $d_i \in D$ has a stronger connection to a topic $t_i \in T$, it is difficult to compare the connection d_i has to t_i and d_i has to another topic $t_k \in T$. Also, the weights for each topic might vary a lot, which might lead to strange keywords extracted from H. For this work, the matrices are always normalised and then, the weights are converted to percentage values to get more consistent, interpretable results. This means that the output rows of matrix W and the columns of matrix H will always sum up to one.

2.2 Topic Modelling on Abstracts

Topic modelling for abstracts provide two main challenges: first, abstracts tend to use specific words which highlight an achievement or describe a problem. As figure 3.6 shows, words like results, data study and based occur in more than a third of all abstracts. The second problem is that the abstracts are rather short and most topic modelling algorithms tend to work better for longer documents. However, abstracts could be a very valuable task because of the very nature of abstracts: they should capture the content of a document without any unnecessary information. Therefore, abstracts should not contain much noise which can be very valuable. There is a case study which shows that the topics extracted from abstracts by LDA can indeed be meaningful [Anupriya and Karpagavalli, 2015]. There are also some approaches which adapt LDA or NMF to fine-tune the algorithms for short texts [Pinto Avendano et al., 2006, Yan et al., 2013], however, these approaches do not guarantee to provide an overall improvement on any data set consisting of abstracts.

2.3 Recent Advances in Natural Language Processing

Due to the recent hype around deep learning due to for example breakthroughs in image classification tasks by using convolutional neural networks (CNN), there are also some recent, interesting advances in NLP. Some of them are discussed below: Word2Vec, which is an improvement over numeric representations of words, Doc2Vec, which is a way to represent documents as vector with numeric values and CNN for text analysis, which could improve the precision over current state-of-the-art algorithms.

Word2Vec. Word2Vec [Mikolov et al., 2013a, Mikolov et al., 2013b] intends to address a problem very specific to NLP: how a particular word can be represented as a vector of numeric values without losing valuable information about the semantic of a word. Before Word2Vec, a very common way of representing a word w was a one-hot encoded vector $\vec{w} = 1 \times |V|$, where all entries for $v_i \neq w$ are set to 0 and for the one entry, where $v_i = w$, the entry is set to 1. However, this one-hot encoded vectors neither do capture much about a word nor are they an efficient way to represent a word since they contain a lot of 0-entries which are not of any use. Alternative ways of creating word embeddings is to use topic models. Each row of the resulting feature-word matrix represents a word. However, topic models tend to be inefficient on very large datasets [Mikolov et al., 2013a] and the word embeddings are corpus specific and can therefore not be used as pre-trained word embeddings for another project.

Word2Vec aims to solve these problems: the word embeddings from Word2Vec can be created in a very efficient manner [Mikolov et al., 2013a] and are also heavily used as pre-trained word embeddings for other projects. The reason this is possible is due to the efficiency of Word2Vec: Word2Vec can be learned on such large corpuses like wikipedia where a large amount of a word's different usages and semantical meanings is covered such that the word embeddings are able to capture the whole semantical meaning of a word accordingly.

There are two different methods, how a word embedding with Word2Vec can be created [Mikolov et al., 2013a]:

- Continuous Bag-of-Words Model (CBOW): With CBOW, the words are predicted given its context. This means that for an input group of words $c = [w_0, w_1, ..., w_r]$, where r is the window size, the output of the neural network will be w_i which is the word which occurs most likely in this context.

- Continuous Skip-gram Model: With skip-gram, the context is predicted given a word. This means that for a input word w_i, the output of the neural network will be the group of words $c = [w_0, w_1, ..., w_r]$, where r is the window size.

The training process works as following: Each word is encoded as a one-hot vector, as described above. Now, if CBOW is used, the one-hot vectors of the context of word w_i forms the input layer. If the skip-gram model is used, the single one-hot vector for w_i forms the input layer. This input layer is then fed into a neural network with one hidden layer and one output layer. For the dimension of the hidden layer, the value

300 is often used in the literature. It defines the number of features a word embedding has. The label follows the same concept as the input layer, but vice-versa: if the chosen underlying algorithm is CBOW, the label consists of the word w_i, otherwise, the output layer will be c. The output layer then is a probability distribution of target words. The task of the neural network is now to maximise the probability distribution of the target words which are in the label and to minimise the probability distribution of the target words which are not. After sufficient iterations, the output layer is discarded and the weights of the hidden layer are used as the word embeddings [Colyer, 2016].

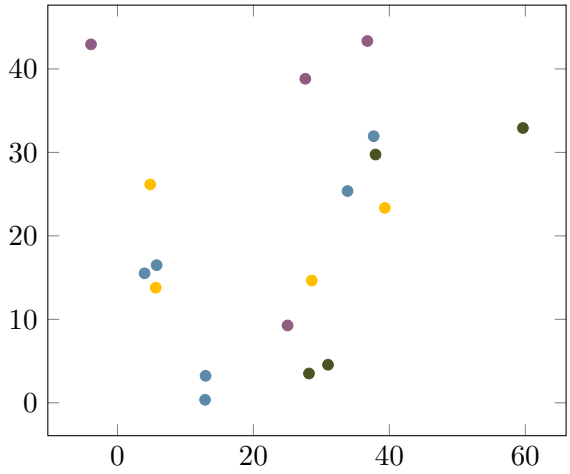
Doc2Vec. Doc2Vec [Le and Mikolov, 2014] is an extension of Word2Vec which aims to provide a vector representation of documents without losing important information like the word ordering, which is lost in an alternative approach called bag of words (BOW).

Doc2Vec adds another vector, a paragraph vector $\vec{p}$, to the input layer of the underlying model chosen for Word2Vec (either CBOW or skip-gram). $\vec{p}$ can be interpreted as representing the context of the document whereas the word embeddings represent the context of the words. The training procedure is similar as the one from Word2Vec except with an additional document context. The result is then a document embedding, having the same dimension as the word embeddings which were trained among the document embedding.

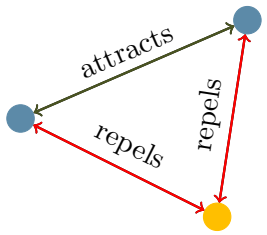
CNNs for Text Processing. Convolutional Neural Networks (CNN) are responsible for major breakthroughs in image classification. Recent research showed, that CNNs can also be applied for certain NLP problems [Lopez and Kalita, 2017]. One downside with CNNs is that the data often needs to be pre-labelled which means that CNNs are not yet particularly suited for document classification (which is mostly unsupervised). The main premise of using CNNs for NLP problems is that these models seem to be able to capture context relevant information. This is due to the convolutional step of the model, which iterates over a certain amount of words (depending on the window size) and therefore captures the context inside this window. The most common task for which CNNs are applied in NLP are currently sentiment analysis. [Kim, 2014] and [Lee and Dernoncourt, 2016] show that CNNs can improve upon the state of the art on some tasks. There is also a case study for sentence classification for health-related text on which they were able to show an improvement of over 15% in terms of accuracy. However, they also had a pre-categorised training data set and therefore, the approach was not unsupervised. [Hughes et al., 2017]

2.4 Evaluation of Topic Models

There does not exist an approach which works for every case to evaluate how well the resulting words from a topic model represent the underlying topic cluster. What is currently used in practice are metrics which measure the semantic similarity of the words which results in the topic coherence. One popular topic coherence metric is the



(a) Initialization of t-SNE



(b) Sample step of t-SNE

Figure 2.3: Concept of t-SNE

Pointwise Mutual Information (PMI) of word pairs [Newman et al., 2010]. The PMI between two words w_i, w_j is calculated the following way:

$$\text{PMI}(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$$

where $p(w_i, w_j)$ measures the probability of seeing word w_i and w_j together in a random document and $p(w_i), p(w_j)$ measures the probability of seeing word w_i/w_j in a random document. The higher the value, the more coherent word w_i and w_j are since they co-occur in the same context more often.

2.5 Visualisation of Topic Models

The topic models in section 2.1 create a (document $\times$ topic)-matrix D. To visualise D in a two-dimensional coordinate system, the rows need to be reduced to a x- and y-coordinate. There are two popular algorithms which can be used for this task: t-SNE and self-organising maps (SOM).

t-SNE. t-SNE [Maaten and Hinton, 2008] is able to project data into a low dimensional space (e.g., a document (document $\times$ topic)-matrix with topic = 20 to a two-dimensional x/y coordinate) without loosing the additional information captured by the higher dimensions. The algorithm has two main steps:

- On the first step, the algorithm projects all the data points randomly onto the lower dimension as can be seen in Figure 2.3a. Points with the same colour were in the same cluster on the original space.

- On the second step, the algorithm iterates a finite number of times through all the points and adapts the position of the points according to its neighbours: neighbours from different cluster repel the point whereas neighbours from the same cluster attract the point as can be seen in Figure 2.3b.

SOM. As t-SNE, SOM [Kohonen, 1990] is able to reduce the data dimension as well as displaying similarities among the projected data by placing them next to each other. SOM is a unsupervised system based on competitive learning, in which the output neurones compete amongst themselves to be activated which results in the neurones organising themselves. The algorithm works the following [Bullinaria, 2004]: Suppose we have n data points.

- The weights of the neurones are initialised to small random values
- A point is randomly picked and the neurone with the weight closest to the point is picked. This is the winning neurone and the neurone is then moved towards the chosen point. Also, n = n - 1 of the neighbour nodes are also moved towards the point, but by a smaller distance.
- This process is repeated until n reaches 0

. As an example, [Bruggmann et al., 2013] used SOM for mapping wikipedia articles into a two dimensional space. They then visualised the resulted clusters by using topographic distortions to highlight very dense clusters.

3

Data Analysis

The data used for this project is provided by Zurich Open Repository and Archive (ZORA)[1], a repository which hosts research material mainly created by people at the university of Zurich. This thesis will only focus on the data which is made available by the ZORA export API, since ZORA is extensively used by the members of the DSI. The data and therefore, the following analysis, is restricted to the members of the DSI.

3.1 Data Quality

As the DSI consists of mainly researches residing in Switzerland, the language the research material was written in might not be English (the national languages of Switzerland are German, French, Italian and Romansh). Therefore, an analysis was made on the language the research material of interest was written in.

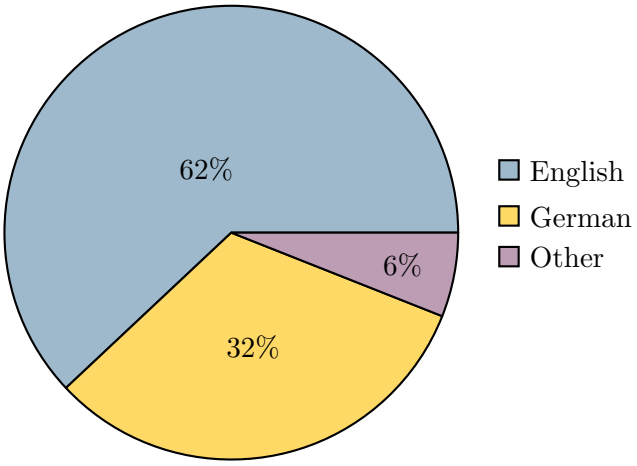


Figure 3.1: Language Distribution on Research Material of Interest

[1] https://www.zora.uzh.ch/

As can be seen in Figure 3.1, two languages, German and English should be taken into consideration as about only two third of all the research material is in English and one third is in German which is a lot, considering the research language which is English.

Since ZORA not only hosts research papers but also other kinds of research material like books and newspaper articles, another question to ask is how many of the research material of interest also provides an abstract which actually can be used for the data processing.

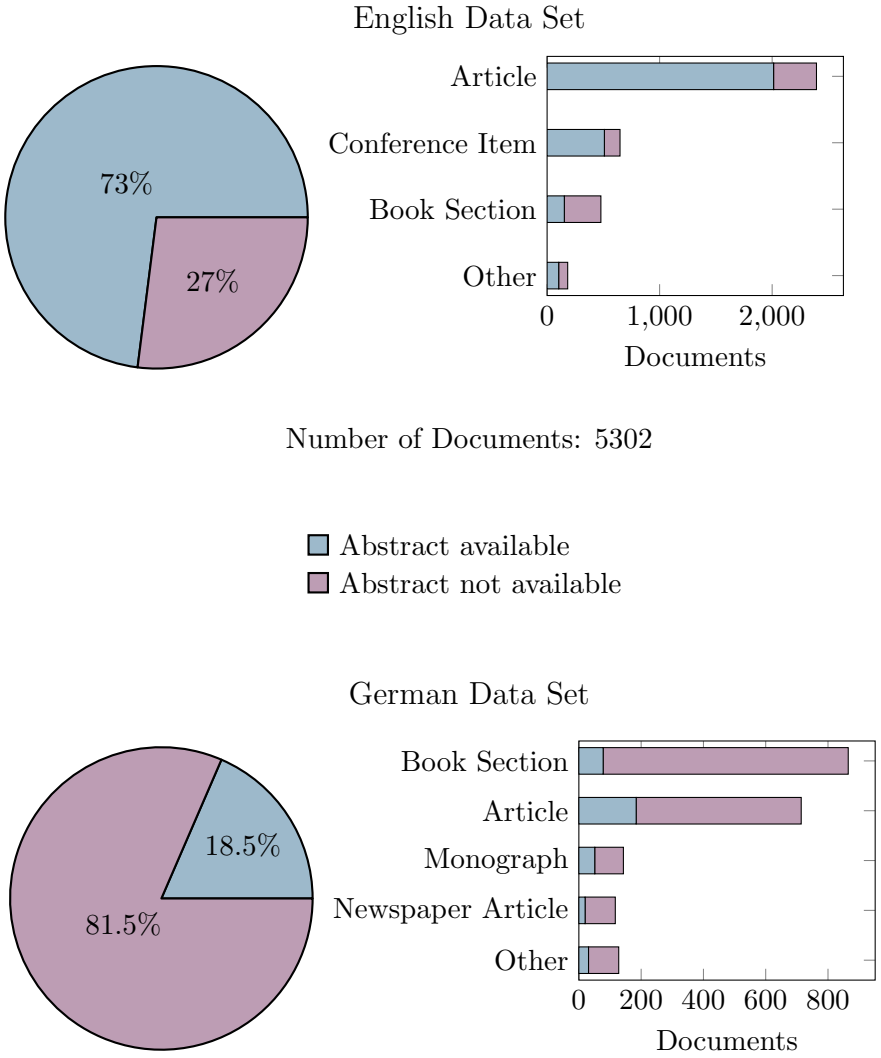


Figure 3.2: Quality of the Data

As can be seen in Figure 3.2, the data quality is mediocre at best. Both, the English as well as the German dataset are missing a lot of abstracts. The German dataset

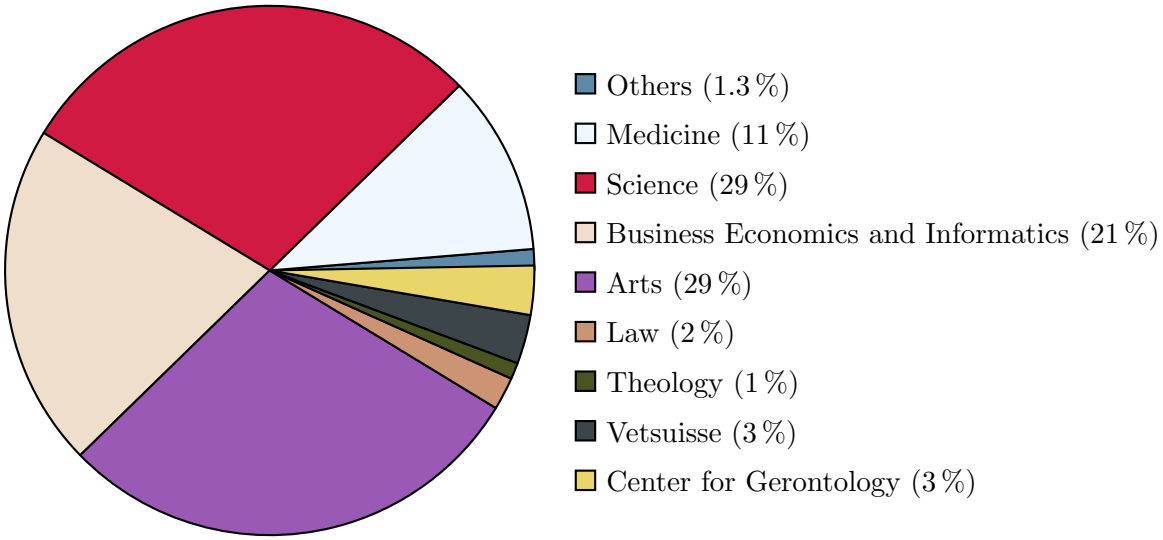


Figure 3.3: Document Distribution among faculties

is providing too few abstracts to be of any use. Therefore, the decision was made to completely ignore the dataset for the time being.

Natural Clusters. What is important for this specific data set is, that there exist natural clusters, the documents may grouped into: the authors faculties and institutes. While these clusters are not of interest for the topic map this work is about, since these clusters are already known, this still might show some insight about potential biases in the data.

Figure 3.3 shows the document distribution among the faculties represented in the data set from the DSI. There is a total of eight faculties presented in the data set, with some people currently not assigned to a faculty ("Others"). As the figure shows, three faculties provide almost 80% of all documents. This means that the topics discovered by the topic model algorithms may be biased towards these faculties.

Figure 3.4 further specifies this bias. In total, there are 38 institutes represented in the data set from the DSI. From these 38, 15 institutes contributed more than 50 documents, the institutes which contributed less are grouped into the "Others"-group to keep the figure readable. There exist three groups which have a substantially higher document count then the others: Geography, which contributed 621 documents, which is about 18% of all the documents, Psychology, which contributed 482 documents, which is about 14% of all the documents and Informatics, which contributed 451 documents, which is about 13% of all the documents. Therefore, three from a total of 38 groups contributed about 45% of all the documents. It is therefore expected, that documents from Psychology, Informatics and Geography dominate the outcome of the topic models.

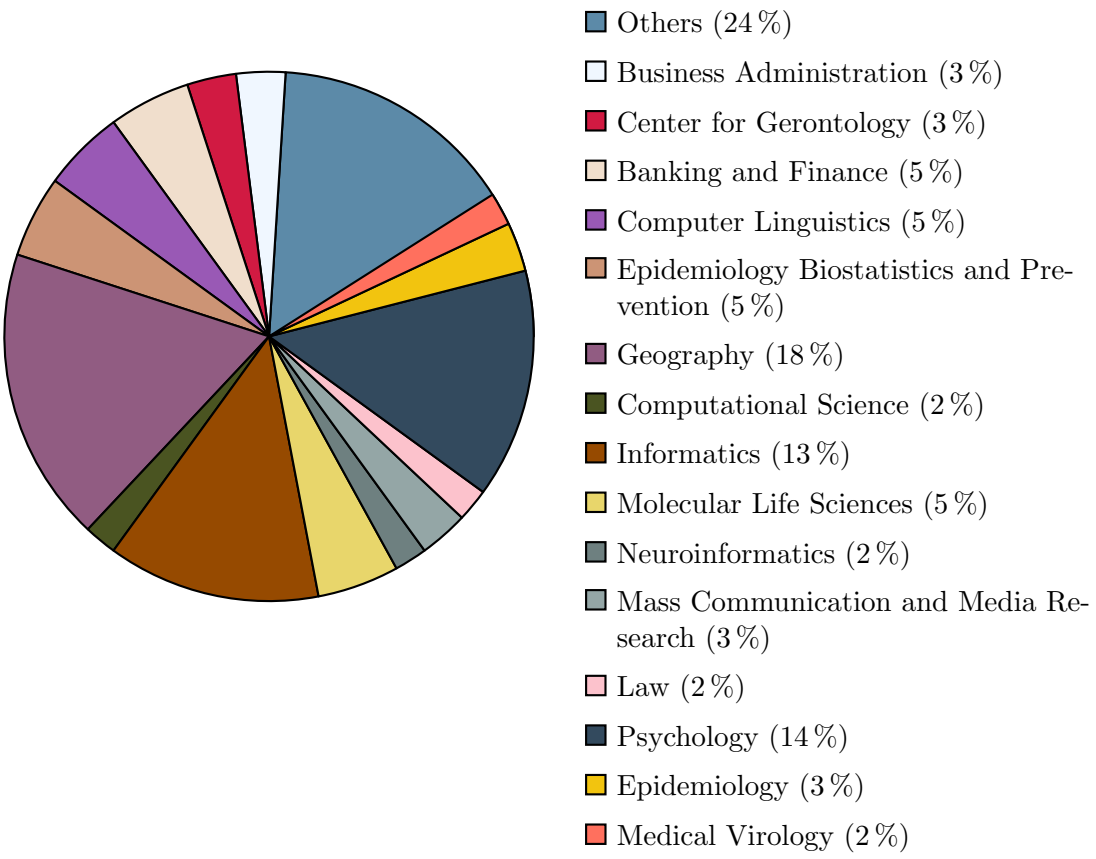


Figure 3.4: Document Distribution among institutes

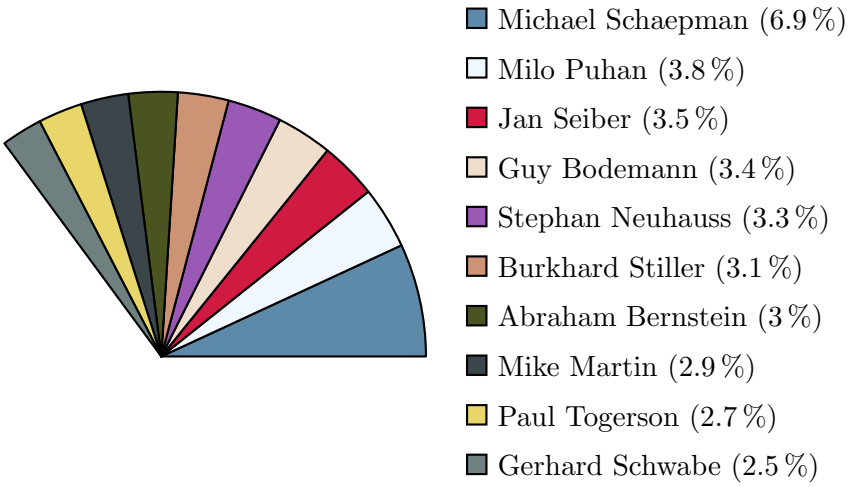


Figure 3.5: Document Distribution among authors

Author Distribution. Figure 3.5 shows the document distribution among the authors. There are currently 181 potential contributors from DSI whose papers are fetched from ZORA. From these 181 people, 134 people have actually contributed some paper to ZORA. The top then contributors are represented in this figure and as can be seen, these authors are contributing about 35% of all documents. Therefore, there might be some topics which are defined by an author, especially, if the author has a special field of interest.

3.2 Preprocessing

There are a lot of information about the research materials provided by the ZORA export API. Of interest for this project are the following informations:

- The id which identifies the document. This id will be used in the database to be able to reference the paper on ZORA. This is currently not used, but may be useful for an improved version of this project.
- The title of the document. The title is used in the visualisation part for the user to be able to identify the papers.
- The abstract of the document (if available). This is the main data source which is used to find the topics the authors wrote about.
- The date the paper was submitted. The date format is not specified and can be arbitrary and is therefore of limited use. The year is the part of the date which is available the most and is therefore parsed and appended to the documents which provide it. The date is used for filters as well as timelines on the visualisation.
- The language the document was written in. However, this information is of limited use. If a document is written in German, this does not mean that the abstract is not available in English. Therefore, the language is not used.
- The authors which wrote the document. However, the names are inconsistent. ZORA seems to map the different versions of the names internally to some extent and provides the correct papers when searched for an author. However, this name mapping is not available in public. It is also not guaranteed that this mapping is always correct. However, it is the best solution available. This names are however not used in the project due to its inconsistency. Instead, the documents are fetched author-wise and to each document, a predefined version of the authors name is assigned to ensure that all documents have the same version of the name.

3.2.1 Removing stop words

For extracting topics, it is crucial to remove stop words. Stop words are words which are extremely common along the documents but do not capture any relevant information

about the context of the documents. There are two type of words which can improve the topic extraction process when removed: language specific stop words and corpus specific stop words.

Language specific Stop Words. As only the english data set is good enough for extracting topics, only english stop words are considered. Examples for words which are removed are the article 'the' or the words 'and' and 'or'. This project uses the stop word list from NLTK (Natural Language Toolkit)[2], a very common toolkit for natural language processing for the programming language python.

Corpus specific Stop Words. One problem natural language processing has particularly with abstracts is that abstracts tend to be written in a similar way with similar words. This means that there are certain words which do normally capture relevant information but since all the documents are written in the same context, the relevant information is lost in this case and therefore, the words should be removed.

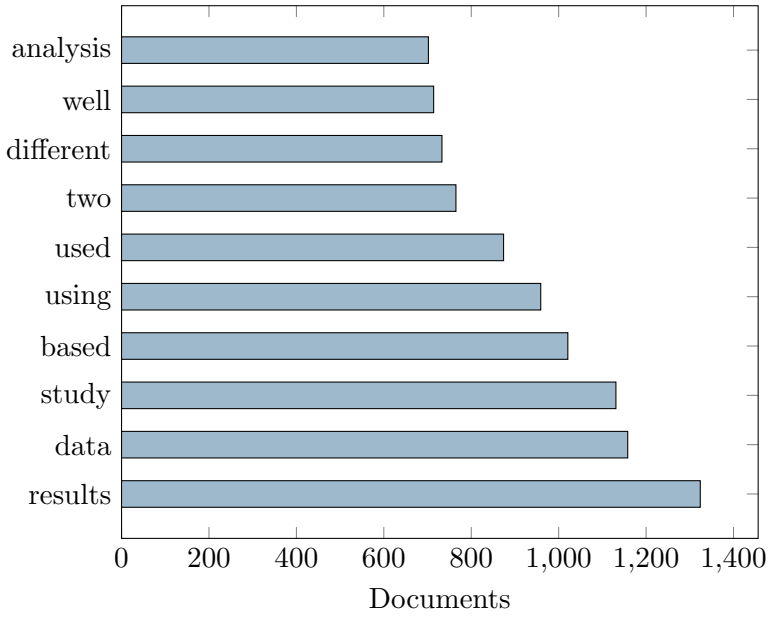


Figure 3.6: Most common words across the documents

Figure 3.6 shows the top ten words which occur in different abstracts. As can be seen, words like results and data occur in about a third of all the abstracts. This tend to mess up the topic models, since abstracts seem to be similar due to having similar words. As mentioned in section 2.1, TF-IDF is a metric which captures the importance of a word, with the intuition that words, which occur in almost every document do not provide much context information and are therefore not of any use. Penalising these words by giving them a really small weight might reduce a lot of noise in the data.

[2]http://www.nltk.org/

Another way (not mutual exclusive to TF-IDF) is to create a corpus dependent stop word list. This is achieved by setting a lower- and upper threshold on the document frequency and exclude all words which exceed or undercut those thresholds.

However, if the words which occur in almost every document are stripped, the topic models might not be able to capture some topics. For example, the word 'data' might be useful to capture topics about data mining and therefore, probably should not be stripped from the data set.

3.2.2 Removing noise

There are also words in the data set which are noisy by themselves and do not capture any information about the context of the document. Examples are links to a website or abbreviations. Removing this words reduces the noise and improves the keywords the topic models might find. In this project, any word which contains non-alphabetic characters as well as any words which consists of less than three characters are removed.

Grouping the same words together. Entries five and six (from the top) from figure 3.6 are very interesting: they are the same word but are present in a different word form. Such words also might add a lot of noise since their information is present more than one time in the data set. This kind of noise can be reduced with stemming. Stemming is a process, originating from linguistic morphology, where all the words are reduced to the corresponding stems. This can be done in different ways: One popular example used in this work is the porter stemmer, where each word is reduced until no more than a certain number of syllables are present. It is important to realise that the resulting stem is not guaranteed to be grammatically correct. However, since this is not used to linguistically analyse the words but only to group the same words together, this is not of any concern.

Removing Content in other Languages. As mentioned above, the language of the abstracts cannot be filtered beforehand and therefore, needs to be done in the preprocessing step to not add any noise. There are two different type of abstracts in this data set: abstracts in a single language and abstracts which are translated in multiple languages. There is no specification available on how the abstracts in the second case should look like and therefore, the language recognition has to be done on a sentence level instead of a document level which increases the preprocessing time heavily. However, to not loose any data as well as to not add noise to the data set, this seems to be worth the additional time. To detect the language, langdetect[3], a python framework based on an open source algorithm provided by Google is used, which promises a precision of over 99%. This algorithm generates language profiles from training corpuses (wikipedia was used) like what accents are specific for which languages, which sequence of characters occur in which language and based on that, are able to analyse the languages of sentences with almost no errors. Therefore, if a sentence is not in english (i.e., another language has

[3]https://github.com/Mimino666/langdetect

a higher probability value of being the main language of the sentence), the sentence is stripped from the text.

4

Topic Model Evaluation

In this section, topic models are evaluated in terms of how well they perform on the dataset analysed in chapter 3. The following topic models are considered in this evaluation: *LDA*, *NMF*, *Doc2Vec + k-Means* and *CNMF*. Whereas the first two topic models are state-of-the-art algorithms for document classification, we developed the latter two for this work with the hope to improve the topic classification on smaller data sets due to the explicit integration of a words context into the models. The sections about the topic models are divided into the following parts:

- Description: For the last two topic models, a detailed description on how they work is provided. For the first two, the reader can consult the corresponding parts in section 2.1 as well as reading the referenced papers and tutorials.
- Preprocessing: For each topic model evaluated, the specific preprocessing procedure is described. This part relies heavily on the information found in Chapter 3.
- Algorithm: Parts of the algorithm are described when necessary (e.g., the initialisation parameters).
- Evaluation: The results are evaluated by using the evaluation techniques from section 4.1

For the topic models, the number of topics is set to 20. This was evaluated by hand after making a grid search through different options. There is also a threshold induced such that only documents with a affiliation of over 5% actually belong to a topic. This value was chosen based on intuition. The chapter concludes with a discussion where the topic models are compared.

4.1 Evaluation Techniques

To evaluate the keywords resulting from the topic models, the following methods are used:

- Coherence Score: To calculate the PMI score, mentioned in section 2.4, an adapted version is used which utilises wikipedia as corpus. The keywords are mapped onto word embeddings from a pre-trained wikipedia corpus. Then, between all possible word embedding pairs, the cosine distance is calculated (as the word embeddings are vectors). From all of these cosine distances, the average is taken which then represents the topic coherence score.

- The top three institutes which contributed papers to the topic are listed and compared to the topic words. While this comparison should be taken with a grain of salt since it is done on a subjective level, it still can provide a solid overview on how well the words capture the underlying documents.

- The document-topic matrices are evaluated on how well they are able to capture the natural clusters of the faculties, talked about in section 3.1. This is done by grouping the documents into faculties. For each faculty big enough (70 or more papers), there will be two groups, one group contains documents which are part of this group and one contains the same amount of documents which are not part of this group. Then, a logistic regression is performed with the corresponding document-topic features to see, if they are able to capture the natural clusters. While the resulting value cannot be used as a general value to indicate how good the topic model is, in this case, where there are three faculties which dominate the data, a high value is expected from a good topic model.

4.2 LDA

Preprocessing. As LDA is a probabilistic topic model [M Blei, 2011], the input matrix is required in the form of integer counts. This means that TF-IDF cannot be used. Instead, a count vectoriser is used which counts the occurrence of each token and uses that count instead of the token value.

Common english stop words are filtered by using NLTK's English stop word list. Also, a corpus specific stop word list is generated by excluding words which occur in more than 75% of the documents as well as words which occur in less than 2 documents.

The words are tokenised by excluding words which do not consist of only letters as well as words which consist of fewer than three characters. Also the words are stemmed by using NLTK's Porter Stemmer to reduce as much noise as possible.

For all of these steps, an evaluation was done by hand and the parameters which seemed to result in the best keywords were chosen.

Algorithm. For LDA, SKLearn's [Pedregosa et al., 2011] implementation of the algorithm is used. The following parameters were chosen by studying SKLearn's user guide [SKLearn, 2017a] as well as evaluating the best possible parameters by hand.

- learning method: the normal batch method is used since the normal batch method seems to produce better results than the online batch method.

Topic ID	Top five keywords					Score
1	using	production	costs	increased	effective	-0.21
2	data	using	approach	based	models	0.12
3	user	services	network	based	algorithm	0.48
4	system	network	using	models	implementation	0.12
5	age	cognitively	performance	tasks	memory	0.35
6	models	using	catchment	flow	data	0.17
7	research	media	communication	designed	social	-0.18
8	using	imaging	data	models	measurements	-0.07
9	brain	effective	studies	cortex	groups	-0.74
10	relationship	couple	cope	stress	studies	0.07
11	infection	gene	transmission	mutations	patients	-0.64
12	using	maps	methods	results	interaction	0.20
13	zebrafish	visualize	development	mutants	mutations	0.24
14	using	development	research	approach	language	0.19
15	patients	treatment	studies	effective	outcomes	-0.16
16	patients	groups	compare	conclusions	methods	0.47
17	estimate	disease	global	regionalized	year	0.20
18	soil	glacier	mass	area	water	-0.06
19	cell	expression	activity	proteins	induced	-1.19
20	measurements	value	differences	oxygen	changing	0.48
						-0.16

Table 4.1: Top 5 Keywords per Topic for LDA

- maximum iterations: The maximum number of iterations before timing out is set to 1500 to ensure that the algorithm has enough time to get the best results possible.

Evaluation. Table 4.1 shows the top five keywords per topic. The score on the right is the coherence score which indicates how well the words can be interpreted by a user, where a high value means that the words are better interpretable. Table 4.2 shows the top three contributors per topic.

As can be seen on table 4.2, a lot of the topics are dominated by Informatics, Psychology or Geography. This is due to the dominance these three institutes have as was indicated in Section 3.1 and seem to lead to keywords which are rather hard to indicate as the score indicates as well. Topics 1-4, 6, 8, 12 and topic 14 are represented by rather generic (at least for abstract specific data sets) words. Compared to Table 4.2, those topics are all dominated by at least two of the three institutes mentioned before which is the cause why these words are so similar.

There are, however, some topics which do have good keywords representing them. Examples are topic 5, which seems to be an age and brain health related topic which is dominated by Psychology and Gerontology, topic 10, which is related to the effect

Topic	Top Contributors		
1	Informatics (72 papers)	Psychology (45 papers)	Section of epidemiology (37 papers)
2	Geography (237 papers)	Informatics (157 papers)	Computational Linguistics (102 papers)
3	Informatics (210 papers)	Geography (124 papers)	Psychology (47 papers)
4	Informatics (109 papers)	Geography (105 papers)	Neuroinformatics (76 papers)
5	Psychology (187 papers)	Center for Gerontology (73 papers)	Informatics (53 papers)
6	Geography (209 papers)	Mass Communication and Media Research (77 papers)	Psychology (60 papers)
7	Informatics (282 papers)	Geography (150 papers)	Psychology (146 papers)
8	Geography (367 papers)	Psychology (60 papers)	Computational Science (25 papers)
9	Psychology (193 papers)	Geography (49 papers)	Dynamics of Healthy Aging (27 papers)
10	Psychology (237 papers)	Medical Virology (50 papers)	Center for Gerontology (32 papers)
11	Molecular Life Sciences (96 papers)	Section of epidemiology (72 papers)	Medical Virology (51 papers)
12	Geography (195 papers)	Informatics (112 papers)	Computational Linguistics (96 papers)
13	Molecular Life Sciences (118 papers)	Psychology (37 papers)	Geography (35 papers)
14	Informatics (224 papers)	Geography (172 papers)	Computational Linguistics (159 papers)
15	Psychology (205 papers)	Epidemiology, Biostatistics and Prevention (142 papers)	Center for Gerontology (48 papers)
16	Epidemiology, Biostatistics and Prevention (75 papers)	Diagnostic and Interventional Radiology (44 papers)	Banking and Finance (40 papers)
17	Geography (95 papers)	Section of epidemiology (52 papers)	Psychology (41 papers)
18	Geography (279 papers)	Computational Science (60 papers)	Psychology (39 papers)
19	Molecular Life Sciences (83 papers)	Psychology (54 papers)	Banking and Finance (47 papers)
20	Psychology (72 papers)	Computational Linguistics (26 papers)	Banking and Finance (23 papers)

Table 4.2: Top Contributors per Topic for LDA

relationship has on people, which is mainly dominated by Psychology and topic 18 which is about climate changes and which is dominated by Geography. The topic with the highest coherence scores is topic 3, and should therefore be the topic which can be interpreted the best next to topic 16. Topic 3 seems to be a topic about distributed services, an informatics specific theme, which correlates with the top contributor of topic 3. Topic 16 is about studies made on people, which also makes sense regarding its top two contributors. The topic with the smallest coherence are topic 19 and topic 9. They do make a lot of sense regarding their contributors, however, labelling the topics with a single category would be a difficult task and the interpretation needs some domain-specific knowledge. Therefore, the coherence score does a good job on capturing the general interpretability of the keywords from LDA.

The evaluation of the document-topic matrix by using the logistic classifier to check if the document-topic matrix can be used to group the papers into the faculties resulted in an accuracy of 81%. This means, that the entries from the topic model matrix were good enough to capture the natural clusters with an accuracy of over 80%.

4.3 NMF

Preprocessing. As NMF is not a probabilistic topic model, TF-IDF, mentioned in section 3.2 can be used to reduce the impact of words which occur in a lot of documents have on the generated topic words.

Also, a corpus specific stop word list is built and used in addition to NLTK's English stop word list. For the corpus specific stop word list, words which occur in more than 75% of all documents as well as words which occur in less than two documents are excluded. These values were manually chosen after iterating and evaluating the results by hand.

The words are also tokenised. Only words consisting of nothing else than letters as well as having a minimum length of three are considered. The tokens are also stemmed with NLTK's Porter Stemmer. The reason for this tokenisation process is to reduce noice in the data. The values were chosen after we tried several parameters and observed that they did not have much effect.

Algorithm. SKLearns [Pedregosa et al., 2011] NMF implementation is used to run the NMF algorithm. The algorithm is initialised with the following parameters after consulting the SKLearn guide [SKLearn, 2017b] as well as evaluating the some of the parameters by hand:

- init: To reduce the instability of the algorithm, which would be introduced by randomly initialising the two matrices W and H, the NNDSVD initialisation variant is used, which is suitable for initialising NMF with sparse matrices [Boutsidis and Gallopoulos, 2008].
- solver: Since the multiplicative update solver cannot update zeros presented in the initialisation, but the NNDSVD initialisation leading to a lot of zero values,

Topic ID	Top five keywords					Score
1	data	using	development	approach	research	0.01
2	patients	treatment	copd	outcomes	trials	-0.36
3	services	advisory	internet	mobile	cloud	0.00
4	cope	dyadic	stress	couple	relationship	0.31
5	circuits	spiking	neuron	network	neural	-0.63
6	text	annotators	corpus	biomedical	syntactic	0.89
7	mutations	gene	chromosomal	genomic	genetic	0.18
8	models	predicting	simulations	parameters	estimate	0.06
9	age	memory	cognitively	training	tasks	0.36
10	media	political	news	communication	online	-0.20
11	glacier	ice	climate	balance	mass	0.39
12	infection	hiv	transmission	parasite	disease	-1.10
13	zebrafish	visualize	larvae	eyes	retinal	0.60
14	dark	matter	galaxy	mass	disc	0.54
15	cell	expression	activity	endothelial	inhibition	-1.45
16	imaging	meris	calibration	spectral	resolution	-0.10
17	catchment	soil	water	flow	stream	-0.47
18	market	asset	price	portfolio	investors	1.13
19	canopy	forest	vegetation	spectral	chris	-0.00
20	aligned	translation	parallel	treebanks	language	0.64
						0.78

Table 4.3: Top 5 Keywords per Topic for NMF

coordinate descent is chosen as solver.

- beta loss: The Frobenius norm is chosen as distance function since this is the most widely used.
- tolerance: Here, the default tolerance of $1e-4$ is used as stopping condition
- max iteration: The maximum number of iterations before timing out is set to 1500 to ensure that the algorithm has enough time to get the best results possible.
- alpha: 0.1 is used as a constant for multiplying the regularisation terms after evaluation by hand.
- l1 ratio: 0.5 is used as regularisation mixing parameter.

Evaluation. Table 4.3 shows the top five keywords for the 20 topics found by NMF. The Score in the third column is the topic coherence score calculated, where higher values mean that the topics are more coherent for people to understand. Table 4.4 show the top three institutes with the amount of documents they contributed to the topic.

Of interest are topics 1-5, 8, 9, 11, 15-17 and 19. All of these topics are dominated by at least one of the faculties which where identified in Section 3.1 as potential dominators.

Topic	Top Contributors		
1	Geography (544 papers)	Informatics (433 papers)	Psychology (428 papers)
2	Psychology (169 papers)	Epidemiology, Biostatistics and Prevention (141 papers)	Banking and Finance (55 papers)
3	Informatics (165 papers)	Mass Communication and Media Research (23 papers)	Geography (22 papers)
4	Psychology (182 papers)	Center for Gerontology (24 papers)	Informatics (11 papers)
5	Informatics (87 papers)	Neuroinformatics (75 papers)	Psychology (53 papers)
6	Computational Linguistics (149 papers)	Informatics (140 papers)	Geography (70 papers)
7	Molecular Life Sciences (106 papers)	Medical Virology (12 papers)	Law (10 papers)
8	Geography (188 papers)	Informatics (78 papers)	Psychology (73 papers)
9	Psychology (191 papers)	Center for Gerontology (72 papers)	Dynamics of Healthy Aging (45 papers)
10	Mass Communication and Media Research (99 papers)	Political Science (44 papers)	Informatics (35 papers)
11	Geography (80 papers)	Psychology (4 papers)	Informatics (4 papers)
12	Section of epidemiology (87 papers)	Medical Virology (55 papers)	Epidemiology, Biostatistics and Prevention (39 papers)
13	Molecular Life Sciences (102 papers)	Geography (43 papers)	Psychology (40 papers)
14	Computational Science (59 papers)	Geography (25 papers)	Psychology (10 papers)
15	Psychology (84 papers)	Molecular Life Sciences (65 papers)	Banking and Finance (46 papers)
16	Geography (226 papers)	Diagnostic and Interventional Radiology (34 papers)	Psychology (26 papers)
17	Geography (193 papers)	Informatics (24 papers)	Psychology (23 papers)
18	Banking and Finance (73 papers)	Informatics (42 papers)	Business Administration (40 papers)
19	Geography (243 papers)	Psychology (17 papers)	Informatics (7 papers)
20	Computational Linguistics (76 papers)	Informatics (36 papers)	Geography (23 papers)

Table 4.4: Top Contributors per Topic for NMF

That by itself might not be a bad thing, however, there are some topics (namely, topic 1 and topic 8, topics which have a lot of papers from all three faculties) which are meaningless for the user and cannot be interpreted without difficulties, even though they may have a high coherence score. Topic 1 and topic 8 appear to have a lot of words in it which are common in a lot of abstracts and therefore, do not provide much meaning.

Also of interest is topic 13, which has a some very specific words as keywords. After a more in-depth look at Stephan Neuhauss, one of the top ten overall contributors (as can be seen in Figure 3.5) documents, 81 of 113 documents written by him (2.4% of all documents) contain 'zebrafish' in its title. Therefore, an author is able to dominate a whole topic by contributing many documents for a specific field of interest.

Topics which seem to be particularly good are topics 6, 7, 9, 11, 12 and 18, which except from topic 12, all have a relatively high coherence score. Topic 6 contains mostly Computer Linguistic specific keywords. Also, a connection to Informatics makes sense, as they do have some overlapping fields of interest. The keywords of Topic 7 also do make sense in connection with the two top contributors, Molecular Life Sciences and Medical Virology. Topic 9 seems to be very specific to ageing and having a healthy brain. All three contributors have a lot to do with these topics. Topic 11 is a geography specific topic which seems to be about climate changes, which matches with the top contributor for this topic. The keywords of Topic 12 matches well with the research fields of Epidemiology and Virology and topic 18 clearly belongs to Banking and Finance as well as Business Administration. It is also interesting that Informatics is heavily represented in this topic since at the University of Zurich, Informatics has a strong tie to economics (it belongs to the same faculty).

The topics with the lowest coherence scores are topic 15 and topic 12. While topic 12 makes a lot of sense in terms of its contributors, topic 15 doesn't seem to make much sense at all. Also, topic 12 is rather easy to interpret, even though the words are very domain specific. This is due to the high impact HIV had in the recent years on people's lives and therefore, the discrepancy between the coherence score and the interpretability.

The evaluation of the document-topic matrix, by using the logistic classifier resulted in an accuracy of 80%. The resulting document-topic matrix therefore seems to be on a par with the document-topic matrix from LDA.

4.4 Doc2Vec + k-Means

The Model. The idea behind this model is very simple: LDA and NMF capture word co-occurrences (among documents), while Doc2Vec considers longer contexts (defined by its window size), which can lead to a richer semantic being captured. Since Doc2Vec is an extension of Word2Vec and therefore, is able to represent a document as a vector, related documents are placed close to each other. This means, that a simple cluster algorithm like k-Means can group together related documents. Then, normal topic model algorithms like NMF or LDA can be used to extract the relevant keywords by running the chosen topic model algorithm for each group. One of the major downsides

of this model is, that if the normal k-Means algorithm is used, a document can only belong to one cluster at a time in contrast to LDA and NMF, where a document can belong to multiple topics at the same time.

Preprocessing. For the first part, creating document vectors and clustering them into n clusters by using k-Means, the only preprocessing done is converting all the words to lower case. There is no need to remove any stop words since then, context information could be lost, as for example a noun should have words like "the" in front of them.

For the second part of the algorithm, extracting the keywords for each cluster, the NMF algorithm from Section 4.3 is used. NMF doesn't need to find the clusters anymore, since k-Means have already done that. Therefore, the number of topics per cluster is set to one (as a complete NMF procedure is invoked for each cluster). However, the document embeddings D are created on a global scale and for each NMF run, the needed rows are extracted. This ensures, that the problems highlighted in chapter 3 do not apply. As an example, for the first cluster, created by k-Means, where documents 2, 5 and 7 are placed into, NMF extracts the most defining words for one topic with document embeddings 2, 5 and 7 as input.

Algorithm. Gensims [Řehůřek and Sojka, 2010] Doc2Vec implementation is used for the Doc2Vec part. Distributed memory is used as the training algorithm with a learning rate of 0.025. The minimal occurrence of words is set to one to not exclude any word, because of the context information which could be lost and the embedding size is set to 300 since that is the size which seems to work best, as it is the most used size in the corresponding literature. The other parameters are set to the default values. This parameter initialisation is mostly based on default values which worked also good in our case.

For clustering, SKLearn's [Pedregosa et al., 2011] k-Means implementation is used. The initialisation method was chosen to be "k-means++", since "k-means++" seemed to lead to the fastest convergence. For the end result, the best of 10 tries is chosen, to get a relatively stable result. Finally, the number of clusters is set to 20, as in all the other algorithms, since 20 topics seems to be a good fit.

The final part of the algorithm is the keyword generation. For the keyword generation, NMF is used with the exact same configurations as in the section above (4.3). The keyword generation works then the following: first, the document embeddings with TF-IDF weights are created as in section 4.3. Then, the algorithm iterates through each cluster and fetches the relevant document embeddings for the documents which are part of the cluster. With this document embeddings as input, NMF extracts the relevant keywords by setting the number of topics to one (as there is one cluster from which we want the keywords from).

Evaluation. Table 4.5 shows the top keywords extracted from each cluster. The score on the right indicates the coherence score, which overall doesn't seem to be that high. The topics which seem to be the most easy to interpret are topic 2, 3, 4, 8, 17, 19

1	catchment	soil	flow	discharge	water	-0.64
2	media	political	news	countries	frame	0.16
3	cope	couple	dyadic	relationships	stress	0.46
4	translation	system	text	treebanks	parallel	1.02
5	imaging	dosing	quality	patient	pitch	-0.10
6	cell	expression	genes	activity	inhibition	-1.09
7	models	data	using	imaging	spectral	-0.07
8	services	user	network	system	designing	0.20
9	measurements	studies	english	age	using	0.27
10	patient	rehabilitation	pulmonary	copd	ventricular	-0.64
11	outcomes	harm	benefit	treatment	alliance	0.22
12	bmp	bone	signal	marrow	cell	-1.02
13	canopy	spectral	forest	reflectance	leaf	-0.15
14	models	data	canopy	imaging	using	-0.07
15	data	models	approach	using	system	0.01
16	system	using	approach	data	user	-0.04
17	stress	relationships	dyadic	couple	cope	0.46
18	spiking	learn	network	neuromorphic	hardware	0.24
19	patient	hiv	copd	disease	exacerbation	-0.99
20	dark	matter	galaxy	mass	clusters	0.32
						-1.44

Table 4.5: Top 5 Keywords per Topic for Doc2Vec

Topic	Top Contributors		
1	Institute of Geography (49 papers)	Department of Psychology (19 papers)	Epidemiology, Biostatistics and Prevention Institute (10 papers)
2	Department of Informatics (43 papers)	Institute of Mass Communication and Media Research (31 papers)	Department of Political Science (21 papers)
3	Department of Psychology (86 papers)	Center for Gerontology (24 papers)	Epidemiology, Biostatistics and Prevention Institute (20 papers)
4	Institute of Computational Linguistics (54 papers)	Institute of Geography (49 papers)	Department of Informatics (44 papers)
5	Institute for Diagnostic and Interventional Radiology (22 papers)	Section of epidemiology (12 papers)	Institute for Law of Nations and foreign Constitutional Law (6 papers)
6	Institute of Molecular Life Sciences (44 papers)	Department of Banking and Finance (26 papers)	Department of Psychology (19 papers)
7	Institute of Geography (81 papers)	Department of Psychology (5 papers)	Department of Informatics (5 papers)
8	Department of Informatics (88 papers)	Institute of Neuroinformatics (18 papers)	Institute of Geography (18 papers)
9	Department of Psychology (61 papers)	Institute of Geography (30 papers)	Department of Informatics (20 papers)
10	Epidemiology, Biostatistics and Prevention Institute (22 papers)	Department of Banking and Finance (21 papers)	Institute of Medical Virology (11 papers)
11	Department of Psychology (22 papers)	Epidemiology, Biostatistics and Prevention Institute (17 papers)	Center for Gerontology (6 papers)
12	Department of Psychology (24 papers)	Institute of Molecular Life Sciences (17 papers)	Department of Business Administration (8 papers)
13	Institute of Geography (69 papers)	Department of Psychology (2 papers)	Institute for Computational Science (2 papers)
14	Institute of Geography (108 papers)	Department of Psychology (4 papers)	Center for Gerontology (1 papers)
15	Department of Informatics (70 papers)	Institute of Geography (66 papers)	Institute of Computational Linguistics (20 papers)
16	Department of Informatics (114 papers)	Institute of Geography (63 papers)	Institute of Computational Linguistics (60 papers)
17	Department of Psychology (93 papers)	Center for Gerontology (17 papers)	Department of Banking and Finance (11 papers)
18	Department of Informatics (26 papers)	Institute of Geography (19 papers)	Institute of Neuroinformatics (9 papers)
19	Epidemiology, Biostatistics and Prevention Institute (41 papers)	Department of Psychology (18 papers)	Institute of Medical Virology (15 papers)
20	Institute of Geography (32 papers)	Department of Psychology (31 papers)	Department of Informatics (26 papers)

Table 4.6: Top Contributors per Topic for Doc2Vec

and 20. In correlation with table 4.6, which shows the top three contributors per topic, topic 2 makes indeed sense as two of three contributors are specialised on media related research. Topic 3 and 17 also makes sense since all three contributors are specified on research about people and their health. Topic 4 is a computer linguistic specific theme, which matches two of the top three contributors. Topic 8 makes sense from both perspectives, the extracted keywords as well as the top contributors, as does topic 19. Topic 20 is also interpretable, as indicated by the coherence score. However, the topic which belongs to Computational Science in the other topic models, is now dominated by the three institutes which dominate the whole dataset, which does not make much sense.

The other topics seem to be rather hard to interpret, which is also indicated by their coherence score as well as the overall coherence score which is rather low.

The evaluation of the document-topic matrix by using the logistic classifier resulted in an accuracy of 66%. The resulting document-topic matrix therefore seems to be a lot lower than the ones from NMF and LDA which means that the natural clusters aren't captured that well by this topic model.

4.5 CNMF

The Model. The goal of this model is to enhance NMF to become context-aware. This would improve NMFs ability to parse abstracts since then, corpus-specific words, as seen in section 3.2 would not be a problem anymore. There is an approach, which enhanced probabilistic matrix factorisation with contextual information by influencing the factorisation with a CNN. The model was used to rating predictions and could outperform other state-of-the-art approaches [Kim et al., 2016]. The following model is inspired by this approach and tries to enhance NMF with contextual information through a CNN.

Overview. Figure 4.1 shows how the algorithm is structured. The algorithm consists of two parts, the NMF and the CNN part.

Both of them have a preprocessing step, which are separated. The reason for this is, that NMF produces better results if noise is reduced by applying stop words and TF-IDF, whereas for CNN context information could be lost if removing to many words from sentences.

The CNN part first fetches pre-trained word embeddings (if not available for a word, the word embedding is randomly initialised) to initialise the Embedding Layer. Then the algorithm steps through the layers until it reaches the output layer which produces a (document $\times$ n)-matrix D, where n is chosen to be the number of topics NMF expects to find. This matrix D is then passed to the NMF part, where it influences the loss function. CNN's own loss function, which is responsible for learning the weights used in all the layers, meanwhile is influenced by NMF's (document-topic)-matrix W.

The NMF part works mostly like the NMF algorithm explained in section 4.3, except for the loss function, which is influenced by the (document $\times$ n)-matrix D.

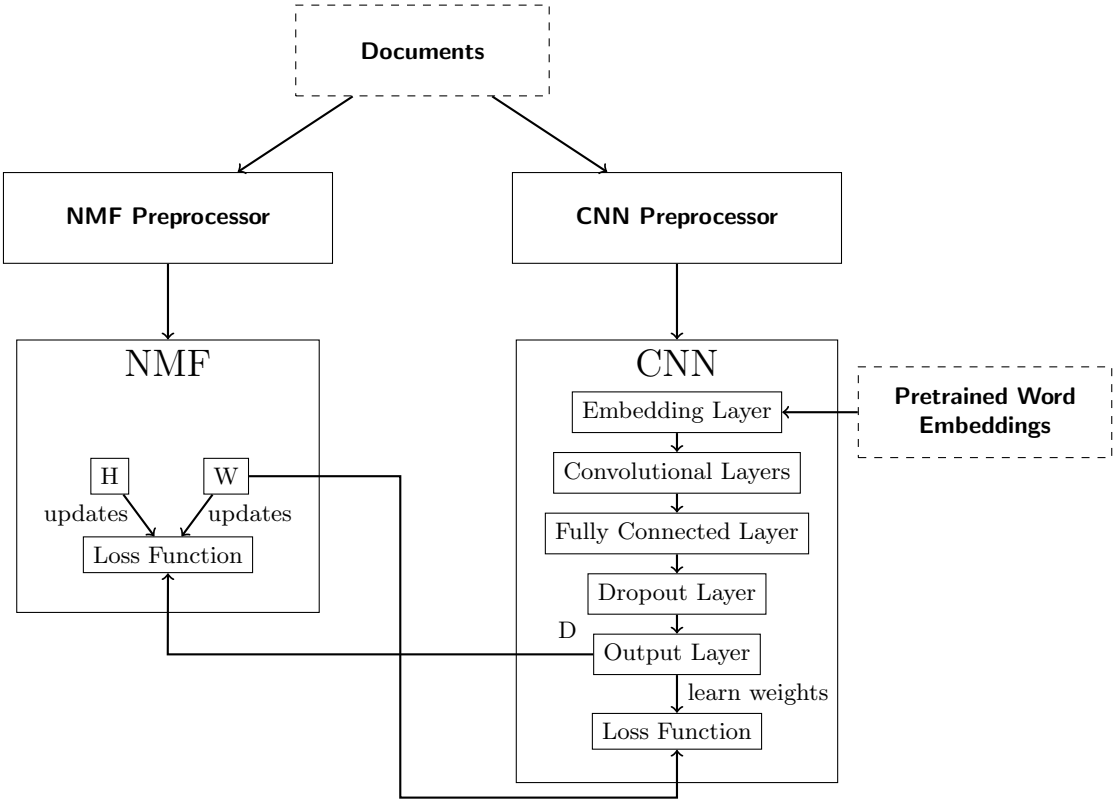


Figure 4.1: Overview of CNMF: A Model which enhances NMF through a CNN

The CNN consists of multiple layers, of which each has a unique task.

Embedding Layer. The goal of the embedding layer is to learn representations of the words used in the documents. There are two possible ways of working with the embedding layer: The first way is to randomly initialise the layer and let the model learn the representations by itself. Another way is to use pre-trained word embedding vectors (e.g. from Word2Vec) and use them to initialise the words. If this approach is chosen, preliminary checks are needed for each word to see if a word embedding exists for them. If yes, the embedding can be used, otherwise, the word embedding for that word has to be provided. Afterwards, the word embeddings can be improved by letting the model adjust the values or they can be fixed. For this model, pre-trained word embeddings over wikipedia, made with GloVe are used.[1] Since about a third of all words to not have a word embedding, the embedding layer still needs to learn and improve the word embeddings for the best possible result and therefore, the word embeddings are not fixed.

Convolutional Layer. The convolution layer is used to extract the features from the text. This is done by creating a filter matrix which extracts a feature map by sliding

[1] https://nlp.stanford.edu/projects/glove/, last accessed on 27 Jan, 2018

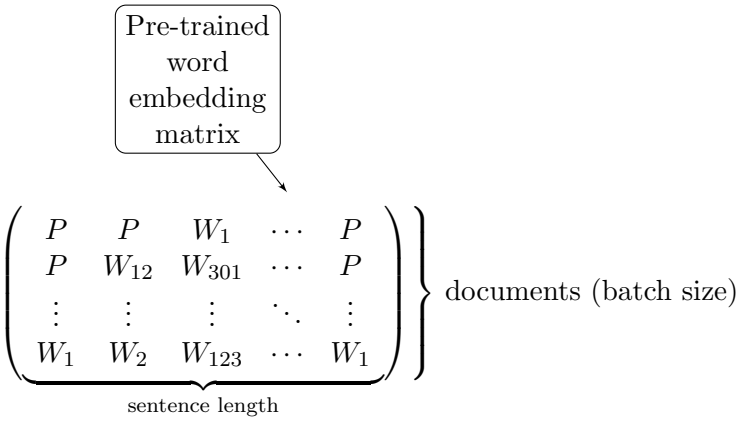


Figure 4.2: Result from Word Embedding Layer

over the word embeddings from each document. On an intuitive level, the filter size corresponds to the window size which defines the upper and lower bounds of words still relevant to define the context of a specific word. Figure 4.3 depicts the convolution step for one document. The green area shows a filter of size (2×3) which means that a total number of $2*6 = 12$ word embedding elements will form a single feature. The feature map is often manipulated by a non-linear activation function. Here, ReLU ($f(x) = max(0, x)$) is chosen, which introduces non-linearity into the model. This is needed due to the fact that most of the real-world data which should be learnt by the CNN would be non-linear [Karn, 2016]. After the ReLU activation, Max-Pooling is applied, which further reduces the feature map to its most relevant components.

Fully Connected Layer. The output of the convolutional step are high-level features of the input documents which now are connected to each other in a fully connected layer to learn additional non-linear features [Karn, 2016]. This is also valuable for a classification task since each feature map would then be connected to all possible classification labels.

Dropout Layer. The dropout layer is used as a regularisation technique to prevent overfitting of the model. What the dropout layer does is to drop visible as well as hidden neurones of the neural network with a defined probability. In this case, these means that the dropout layer randomly drops some of the feature maps.

Output Layer. The output layer is the final point of the CNN which merges everything together to a (document $\times$ n)-matrix where n is the number of captures features. In this case, n will be equal to the number of topics. The output layer is again a fully connected layer, which reduces the output dimension to the number of latent features defined. Afterwards, another ReLU activation is performed.

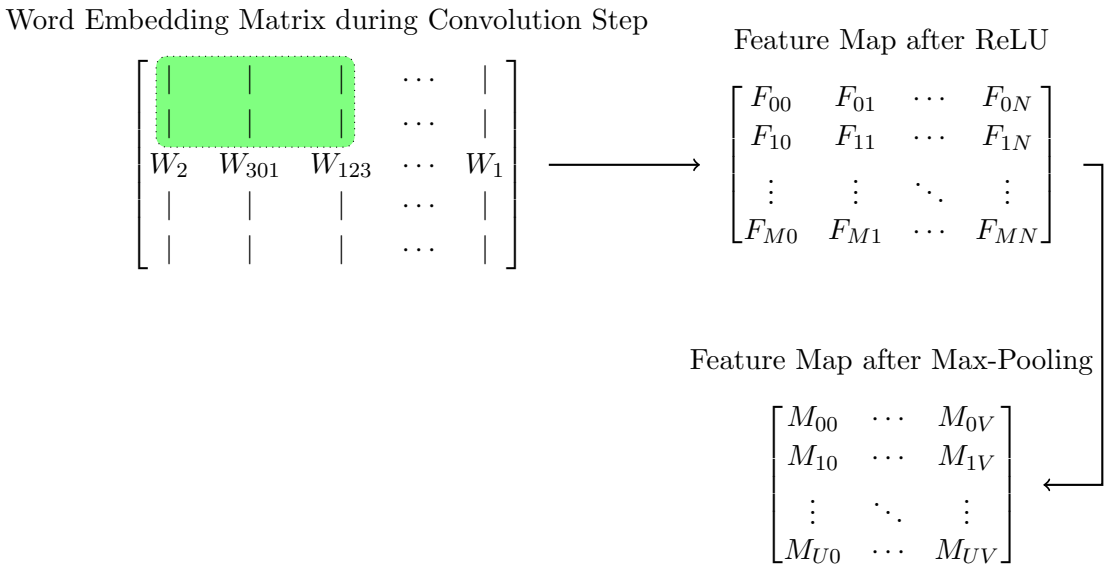


Figure 4.3: Convolutional Layer

The whole Algorithm. The NMF part is preprocessed the same way as the NMF algorithm in section 4.3 except for the two constants λ_1 which weights the impact, the CNN part has on NMF and λ_2 which is a regularisation constant. However, the NMF part does currently not contain any metrics which check if the algorithm converges and therefore stop it early. The NMF loss function is defined the following way:

$$\text{loss} = (A - WH)^2 + \lambda_1 * \text{l2 loss}((W_n - D_n)^2) + \lambda_2 * (\text{l2 loss}(H_n) + \text{l2 loss}(W_n) + \text{l2 loss}(D_n)) \tag{4.1}$$

The first term of equation 4.1, $(A - WH)^2$ is the normal distance the approximation has to the original matrix which is used in the typical NMF algorithm.

The second term, $\lambda_1 * \text{l2 loss}((W_n - D_n)^2)$, is the interesting part of this loss function: this is the place where the algorithm implements the context information by trying to approximate W_n with D_n, where W is the l2-normalised document-topic matrix from NMF and D is the l2-normalised document-n matrix from CNN. The normalisation is necessary since the feature values of both matrices may be too far apart from each other.

The third term, $\lambda_2 * (\text{l2 loss}(H_n) + \text{l2 loss}(W_n) + \text{l2 loss}(D_n)))$, is used to regulate the loss function.

The optimiser used to optimise this loss function is Tensorflow's [Abadi et al., 2015] AdagradOptimizer.

The documents for the CNN part are processed the following way:

The words are tokenised and all words, which contain less than two characters as well as words which do not consist of only letters are excluded. Then, for each token, a unique id is generated and from this, a document-id matrix is created. Then, the word embeddings are fetched from the pre-trained GloVe model and mapped to the corresponding ids. This enables the algorithm to fetch the word embeddings efficiently by using the documents

as a lookup matrix for the word embeddings. Also, the documents are padded to the length of the document with the maximum number of words. This is necessary since CNN can only work with input data of the same length.

The CNN is then initialised with the following parameters:

- Embedding size: Since the CNN part uses pre-trained word embeddings, the embedding size for the embedding layer has to match the size of the pre-trained word embeddings which is 300 for the GloVe embeddings.
- Number of Classes: This decides the number of feature dimensions per document. Since the resulting matrix D from CNN needs to have the same dimension as the matrix W from NMF, the number of classes has to be set accordingly.
- Vocabulary Size: This is needed for the word embedding matrix, which has the dimension (vocabulary size $\times$ embedding size). The word embedding matrix is used to fetch the word embeddings and replace the ids of the words in the input documents with the corresponding embeddings. This equals to the number of unique word ids.
- Sequence Length: The number of words in a document (all of the documents are padded to the same length).
- Learning Rate: The learning rate for the optimiser. 0.01 was found to be the best learning rate for the optimiser to use after evaluating.
- Regularisation constant (λ_2): This regularisation parameter is explained in the loss function further down.
- Weight constant (λ_1): This controls the influence, the distance matrix W has to matrix D.
- Dropout Probability: This is set to 30%, as this seems to lead to the best results.

The layers the CNN use are the following:

- Two convolutional layers. The first one has 32 filters of size (5×5), the second one has 64 filters of size (3×3).
- Both of the convolutional layers are followed by a max-pooling layer with size (2×2), each of which having a stride of 2. This means, that each subregion extracted by the filter should be separated by 2 in both width and height.
- Then, all of the resulting filter maps are flattened to a one-dimensional layer and then connected to a fully connected layer with dimension (flattened layer $\times$ 100).
- Afterwards, a dropout layer with dropout probability of 30% is applied to the fully connected layer. A ReLU-activation is performed on the result.

- Finally, the output matrix D is created inside the output layer by creating another fully connected layer which reduces the output dimension to the number of latent features (here 20). Again, a ReLU-activation is performed on the result.

The loss function of the CNN is the following:

$$\lambda_1 * \text{l2 loss}((W_n - D_n)^2) + \lambda_2 * (\text{l2 loss}(D) + \text{l2 loss}(W)) \tag{4.2}$$

The first term of Equation 4.2, $\lambda_1 * (W_n - D_n)^2$ penalises the algorithm if the distance from the l2 normalised matrix W is too far from the l2 normalised version of CNNs output matrix D. Again, the matrices need to be normalised to be able to calculate the distance. As in the NMF part, the second term, $\lambda_2 * (\text{l2 loss}(D) + \text{l2 loss}(W))$, regularises the loss function.

To optimise the loss function, the same optimiser, Adagrad, as in the NMF part, is used.

Connecting the NMF part with the CNN part First, a complete CNN iterations is run. To avoid out of memory crashes, the CNN iterations have to be done in batches. It is ensured that only the related parts of W and D (the rows which corresponds to the documents used in the batch) are considered for the loss function. The resulting matrix D of a batch has dimension (batch size $\times$ n) and therefore, the results from all the batches are concatenated. Once a complete CNN iteration has finished processing, two iterations of NMF are done: The first one fixes the matrix W and only updates the matrix H with the result from CNN and the other iteration fixes the matrix H and only updates the matrix W with the result from CNN. This is considered one complete iteration of CNMF and is done for a fixed number of times. There are at least 80 iterations needed to see some first results. However, the algorithm is in a stable state only after not less than 300 iterations.

Evaluation. The CNMF model isn't yet ready for an evaluation, since the model doesn't work as it should yet. While on paper, the loss functions make sense, they do not work correctly in practise. The NMF part is very sensitive to the changes made. For a lot of parameters tested $(0.0001, 0.001, 0.01, 0.1, 0.5, 1, 5, 50)$ for the weighting constant λ_1 and the regularisation constant λ_2, the loss function stays in the same range if tested with about 50 iterations. Almost every time, the loss function decreases for a bit, then increases again and later decreases, as if it would have hit a local minima. If the algorithm is run for a few hundred iterations with parameters small enough that the loss function keeps decreasing, the NMF part seems to completely control the output of the algorithm as the keywords are nearly identical to the ones NMF gets if run without the influence of CNN. This is due to the fact, that the original part of NMF's loss function, $(A - WH)^2$, is not weighted and gains more impact the smaller the two constants are. However, if the document-topic matrix is evaluated the same way as for the other algorithms to see if it captures the natural clusters, the percentage is always below 50%, which is worse than a random guess to see if a document to a group or not. The loss function of the CNN

part doesn't have that problem of being unstable. It decreases for almost all parameter values mentioned above.

This can now mean two things: either the CNN part learns something which is completely wrong or the loss function of the NMF needs to be adapted in some way. It may also be that the initialisation method NNDSVD doesn't work for this mixed type of algorithm. The NMF algorithm converges rather fast to a point with NNDSVD which results in rather good keywords, which is the main reason it was used. However, maybe there is another local minimum which results in better words. While some tests were conducted with the NMF part being randomly initialised, without satisfying results, most of the tests were made with the NNDSVD initialisation, since the keywords extracted with a NNDSVD initialised NMF version were much better. It may also be that the CNN structure doesn't work. As an example, ReLU might be an activation function which does not work on word embeddings. However, the CNN structure is based upon the work of [Kim et al., 2016] where they had success with using ReLU as an activation function. Another potential improvement is to fix the sentence length to a value of about 200-300 words. If the maximum number of words is used, a lot of documents might have similar context information due to the padding if there is one big outlier. The main problem of finding the source of error is, that one complete test with about 300 iterations takes more than ten hours on a PC with a rather new graphic card. On a cluster without access to a graphic card, the algorithm most likely times out before finishing. This means, that since there are many parameters used in this model, a grid search for the best parameters might take about a week. Therefore, it may be valuable to improve the structure of the algorithm first to be able to run on a distributed environment to at least save some time.

4.6 Discussion

All of the topic models tested on the data set restricted to abstracts from papers by member of the DSI were evaluated by the following three criteria:

- The coherence score, which indicates how interpretable the keywords are. While the coherence score isn't always able to capture the intuition, as there are some topics which are rather good interpretable by humans but still have a low coherence score, it did a good job on our extracted results, as the idea of interpretable topics where mostly on par with the coherence score. The coherence score also provides a good overview of the overall topic qualities of the models.
- The keywords were evaluated in correlation to the top contributors per topic. Since this evaluation strategy would discard topics with unexpected correlations between institutes, the top contributors were evaluated in correlation to the top five keywords, which seems to be a rather good combination to evaluate the topic models. Again, the clear winner here seems to be NMF, however, all of the topic models seemed to capture the natural clusters rather well, as the third evaluation criteria shows.

- The document topic matrices were evaluated to see if they capture the natural clusters. LDA and NMF seem to capture the natural clusters very well, whereas the Doc2Vec models does a better job than a random guess but is still missing about 25% of the cases

While this evaluations give a good overview about the topic model results, it is not possible to choose a topic model without being objective. For the authors, the NMF topic models was primarily chosen because the coherence score is high and the topics also are the most interpretable on a human evaluation. The top contributors did make sense in all of the topic models most of the time, so this was not a driving factor. Therefore, the NMF model is chosen to be the model of choice for this specific dataset.

5

Visualisation

Mapping documents to a two-dimensional coordinate system is key for visualisation. Everything revolves around documents: authors, topics, institutes and faculties are just groups of documents with a certain label assigned. Connections are also based on the position of documents. The task which need to be done is to reduce the dimension T, which is the number of topics, of document vectors $\vec{d} \in D$ to a two-dimensional coordinate:

$$f(d) : \vec{d} \in \mathbb{R}^T \mapsto \vec{d} \in \mathbb{R}^2$$

There are two algorithms which are commonly used for mapping an element of a higher space to an element in a lower space: SOM and t-SNE, as mentioned in section 2.5.

Section 5.1 will evaluate the two algorithms SOM and t-SNE, based on the data set mentioned in chapter 3. Section 5.2 will explain the algorithm which was used to get a more interpretable label for each topic. Then, section 5.3 will guide the reader through the web application, highlighting the specific choices made to give a better intuition about the extracted data from chapter 4. The web application can be accessed at: *https://rebrand.ly/dsi_topic_extractor*

5.1 Reducing Dimensions

As mentioned in section 2.5, SOM uses a neural network which utilises competitive learning to learn to map an element from a higher space to a lower space, whereas t-SNE uses the neighbours labels to influence the position. As input, the entries from the document-topic matrix D is used which is the result from the topic modelling process. Each entry of a row in the D represents a latent factor of the document. Here, this means how much the document belong to a certain topic (more on that in section 2.1 and chapter 4). Intuitively, it makes sense to group papers together which have a similar topic distribution and therefore, this will be the metric used for the reduction process to place the documents on a two-dimensional coordinate system in relation to each other. For the following evaluation, the results from the NMF topic model, described in section 2.1.2, were used. However, they do not correspond to the version currently used on the web application, as for this Figure, a smaller learning rate was used on a different NMF run. This means, that the topic distribution might be slightly different. The influence

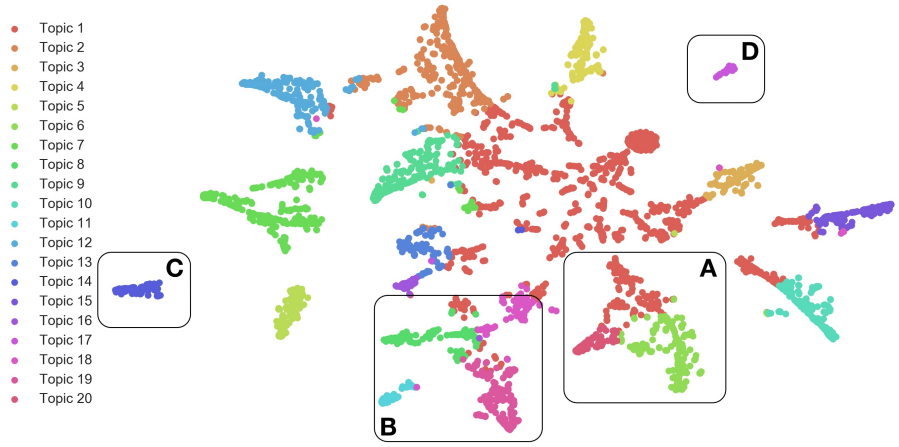


Figure 5.1: The paper distribution created by t-SNE, labelled after the topic they contribute the most.

the learning rate has on the visualisation is how close together the whole scatter chart will be. For this visualisation, a smaller learning rate achieves a better interpretability, whereas for the topic map, a higher learning rate achieves a better interpretability as otherwise, the hexagons would not be connected.

t-SNE. t-SNE is explained in detail in section 2.5, therefore, this part only focusses on how the algorithm was applied to the data and what the results were. To perform the algorithm on the data set, SKLearn's [Pedregosa et al., 2011] implementation of t-SNE is used.

Figure 5.1 shows the result from mapping the document-topic entries into a two-dimensional space. The topic description to the corresponding number can be found in table 4.3. Since the goal of the topic model is to capture intrinsic relationships among documents and the data set is sufficiently large to not know what most documents are about, it is difficult to evaluate if the clusters make sense. However, a few sectors are analysed according to the topics contributors.

Section A groups together three topics: parts of topic 1, topic 6 and topic 20. As can be seen in tables 4.3 and 4.4, topic 1 is a very mixed, general topic. This can also be seen in the topics distribution: it is centred and contains the most papers. The main influencer of topic 1 are Geography, Informatics and Psychology. Topic 6 and Topic 20 are mainly Computer Linguistic topics, but also contain influence from Informatics. Therefore, all three topics are influenced by Informatics and are related to Computer Linguistics, which seem to make sense.

Section B groups together four topics: topic 8, topic 11, topic 18 and topic 19, where topics 8, 18 and 19 seem to be stronger related with each other than with topic 11. All of these topics are heavily dominated by Geography. Topic 18 could be about using models to predict changes, whereas topic 19 is about vegetation, topic 11 is about climate changes and topic 8 seems to be about changes related to water. Therefore, these topics clearly belong together.

Section C only captures one topic, topic 14, which seems to be related to astronomy. This topic is dominated by the institute for Computational Science with some geographic influence. Since they are no physics institute or anything in that direction in this selected data set from the DSI, it makes sense that topic 14 has no neighbours.

Section D also only contains one isolated topic, topic 17. Topic 17 is mainly influenced by psychology and math, however, the key words are really hard to interpret. It does however makes sense that something math related would be a bit separated from the other topics, since most of the influencer seem not to be that math related.

SOM. As mentioned in 2.5, SOM is another algorithm which is able to project data from a higher dimension to a lower dimension. However, since t-SNE already provided a good, interpretable topic map, SOM was not further evaluated for this work.

5.2 Topic Labelling

Inspired by [Kataria et al., 2011], the topic labels are recommended by analysing wikipedia categories and fetching the category which is the closest to the extracted keywords from the topic model described in chapter 4. The idea behind this approach is that wikipedia categories tend to be good general topics. Also, since there are so many wikipedia categories, the chance to find a category which describes the extracted topics in a more interpretable way should be high.

The algorithm works as following: For each of the top ten extracted keywords of a topic, the Word2Vec embedding from a pre-trained wikipedia corpus is fetched. Then, the algorithm iterates through all the wikipedia categories and calculates the distance to each of the keywords. Since categories tend to consist of multiple words, the categories are split and stopwords are removed. Then, the average of the distances from each of the category parts to all of the keywords represents the score a category gets. Sadly, the topic with the highest score tend not to be the best one and we weren't able to find a metric which results in the best topic. One possibility is to weight the keywords according to its position since the higher the keywords are positioned, the more they define the topic. However, no weighting factor which resulted in significant improvements could be found. The best topic category is usually in the top ten entries but has to picked manually. Nevertheless, some of the proposed categories are pretty good recommendations for labelling the clusters.

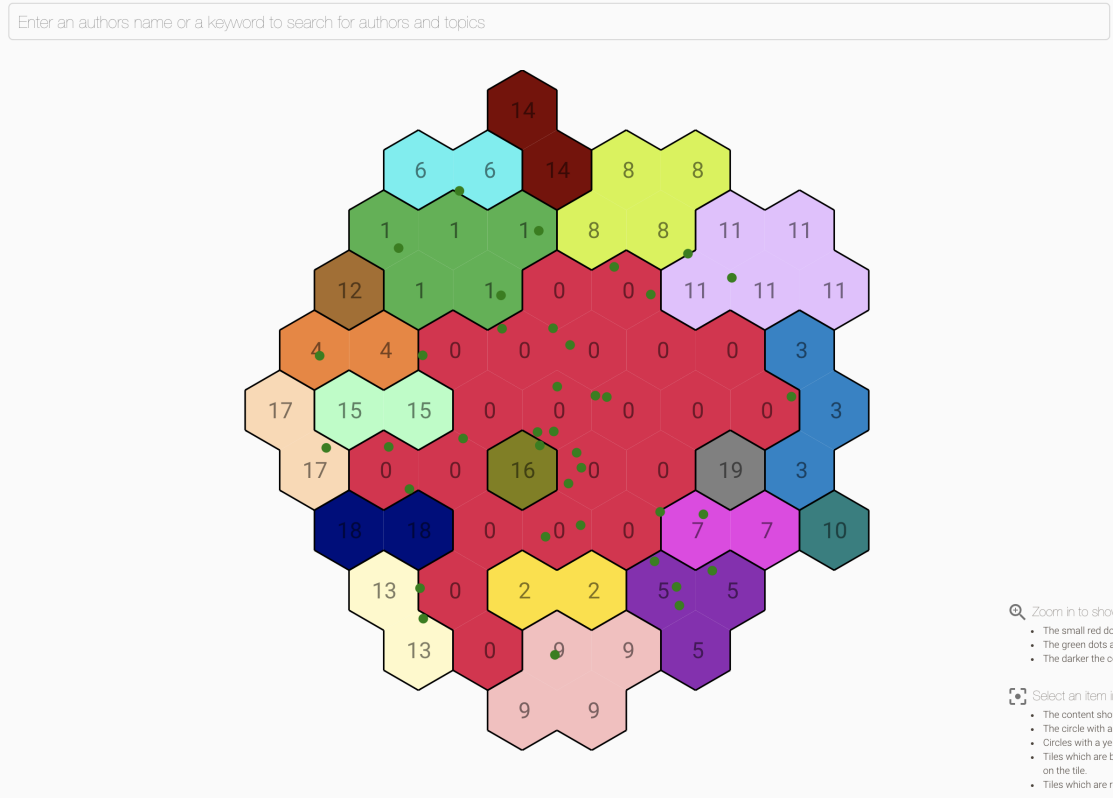


Figure 5.2: The main page. The topic map shows a global overview about the topics. The green dots are some of the authors which contributed the most documents.

5.3 Web Application

The web application consists of two parts: the main part is a topic map, which highlights the relations topics have to each other. The focus of the map can be changed by using the search field and choosing a value to filter the map with. The other important part of the web application are the detail pages: each of the elements of interest, the authors, the institutes, the faculties and the topics have a detail page, where charts help to interpret the extracted data further. As each document can contribute to multiple topics, for the document to be considered belonging to a topic, a lower threshold of 5% was introduced. This means, that a document has to belong to a certain topic by at least 5% to be considered part of a topic. It is important to notice, that the information on the screenshots might not correlation with the data shown in Chapter 4 and the data shown on the live web application, as the web application still was undergoing some changes while writing the thesis. However, the charts, the features shown and the meaning behind the informations are the same.

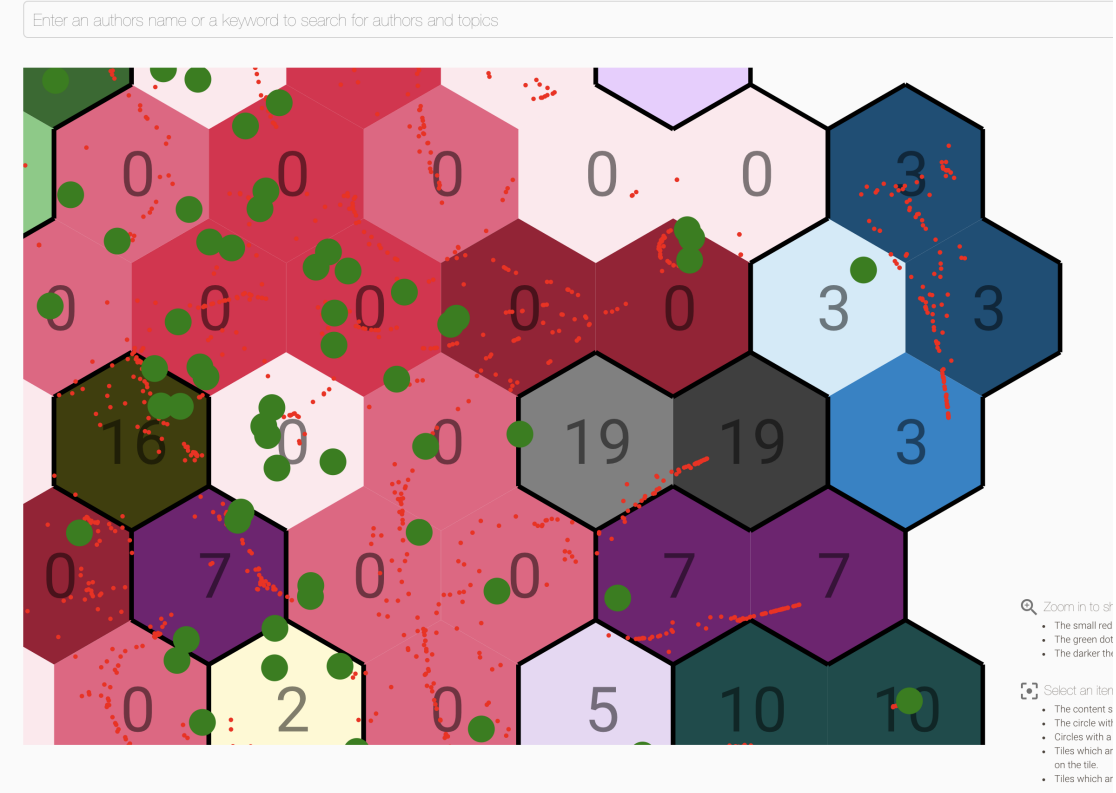


Figure 5.3: The main page zoomed in to reveal more of the structure. All of the authors (green dots) as well as the papers (red dots) are shown

The Main Page. The main page consists of two components: the search field and the topic map. The topic map, which is inspired by [Bruggmann et al., 2013], has two levels. On the overview level, which is shown in Figure 5.2, the topic map shows a compact map of the topics and how they relate to each other. Also, some of the overall top contributors are marked by green circles (at this level, only authors which contributed 25 documents or more are shown). By clicking on a circle, the corresponding detail page of the author is opened. The topic map is also able to show some information about the structure of the topics by zooming in on the map. After a certain zoom threshold is passed, the density of documents for each hexagon is reduced and the hexagons are coloured to represent the number of documents they contain. The darker the colour, the more documents are represented by a hexagon. The map now also shows all of the authors as well as all of the papers on their corresponding places, which is shown in 5.3.

The process of merging a certain amount of points into hexagons is called hexagonal binning. For the hexagonal binning a d3 plugin[1] was used. For the author's position, the mean value of all his documents was chosen, as this represents the position an average document would have on the map.

The search field lets the user search for an authors name, for faculties, institutes and topics. The author can be searched by the last name, first name or a mix of both. The institutes and faculties can be searched by the full name or its abbreviation. The topics can be searched by the topic label or the extracted keywords. If an author, institute or faculty is selected, the topic map will be filtered accordingly: only the related authors (if filtered by an author, the related authors are the author itself and his coauthors; if filtered by an institute or faculty, the related authors are its members) as well as only the documents contributed by the related authors are shown. The map highlights this focus on a specific group of people by changing the colour scheme to a heat map, where blue means few to none documents (the darker the colour the less documents on a hexagon) and red means many to a lot of documents (the darker the colour, the more documents on a hexagon), which is shown in Figure 5.4.

The Author Page. The author page is shown in Figure 5.5 for the author Gerold Schneider. He is part of the Department of Informatics.

On the top of the page, a line chart shows the topic trends. There may be a lot of topics an author has contributed to and therefore, each topic can be deactivated for the line chart by clicking on the corresponding item of the legend on the bottom of the chart.

On the bottom left is a donut chart, which shows the overall number of papers the author has contributed as well as the corresponding topics, the author has contributed to.

On the bottom right is a graph which shows the connections the author has. Each node of the graph represents an author, which is coloured with the institutes colour he's part of. By hovering over the node, the authors name and institute is shown. Also, the user can jump to the authors detail page by clicking on the node. On 5.5, two of the nodes in the connection graph are highlighted by a red border. This means that they

[1] https://github.com/d3/d3-hexbin, last accessed on Jan 27, 2018

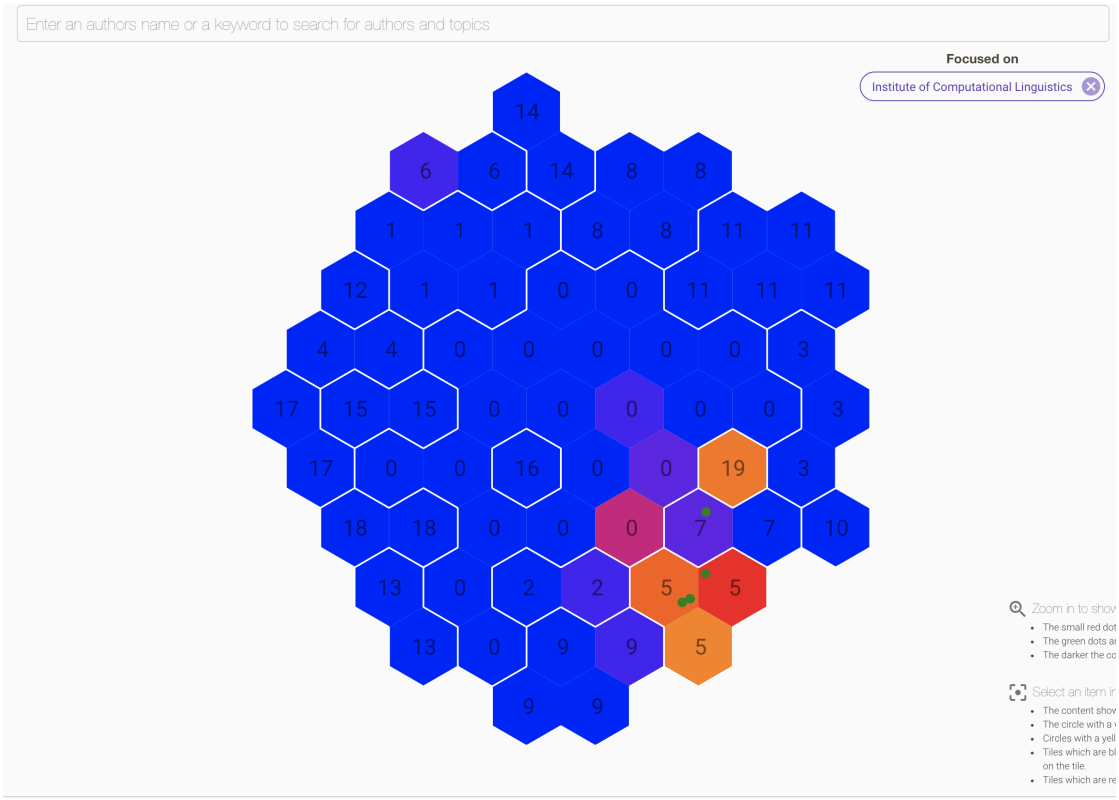


Figure 5.4: The main page focused on the Institute for Computer Linguistics. The tiles where a lot of papers are placed are red, the ones with only a few or none papers are blue. Also, only the contributors belonging to the institute are shown.

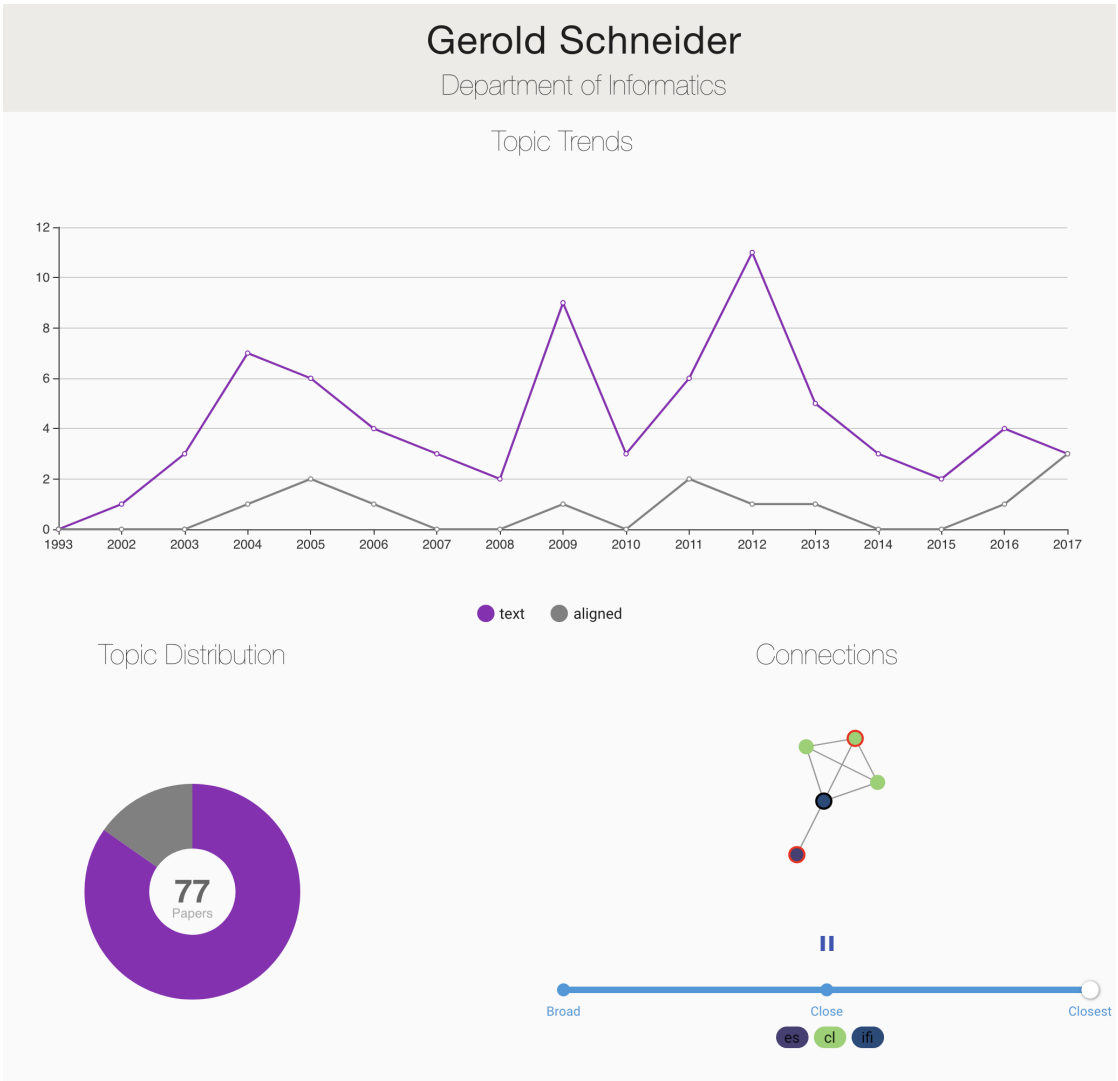


Figure 5.5: The detail page of author Gerold Schneider. The page consists of interactive charts which make the data specific to the author more interpretable.

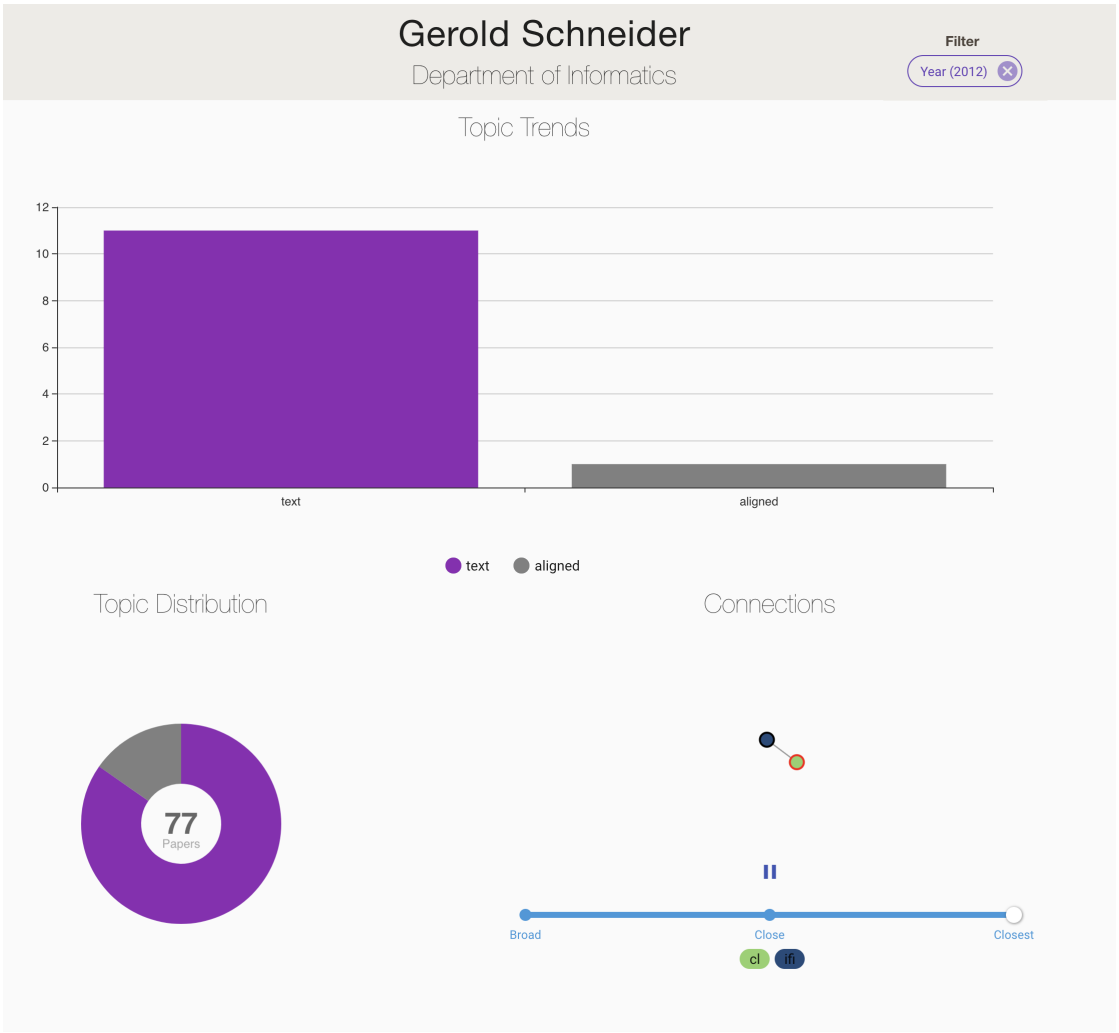


Figure 5.6: The detail page of an author, filtered by the year 2012.

have coauthored at least one paper with Gerold Schneider. The neighbours of an author are calculated by getting the average position a document of the author has (which is equal to the average position of the author itself) and checking the distance to each of the other authors against a certain threshold. For closest, the threshold is 2, for close, the threshold is 4 and for broad the threshold is 6. This values were chosen by hand after evaluating possible values. With the solution currently implemented, it may be that an author has no connections. Another possible approach would be to instead limit the amount of connections shown, e.g., for closest, show the three closest authors.

All of the described charts can be filtered by a certain year. If the user clicks on a year on the timeline, only data related to the selected year is shown, as 5.6 shows.

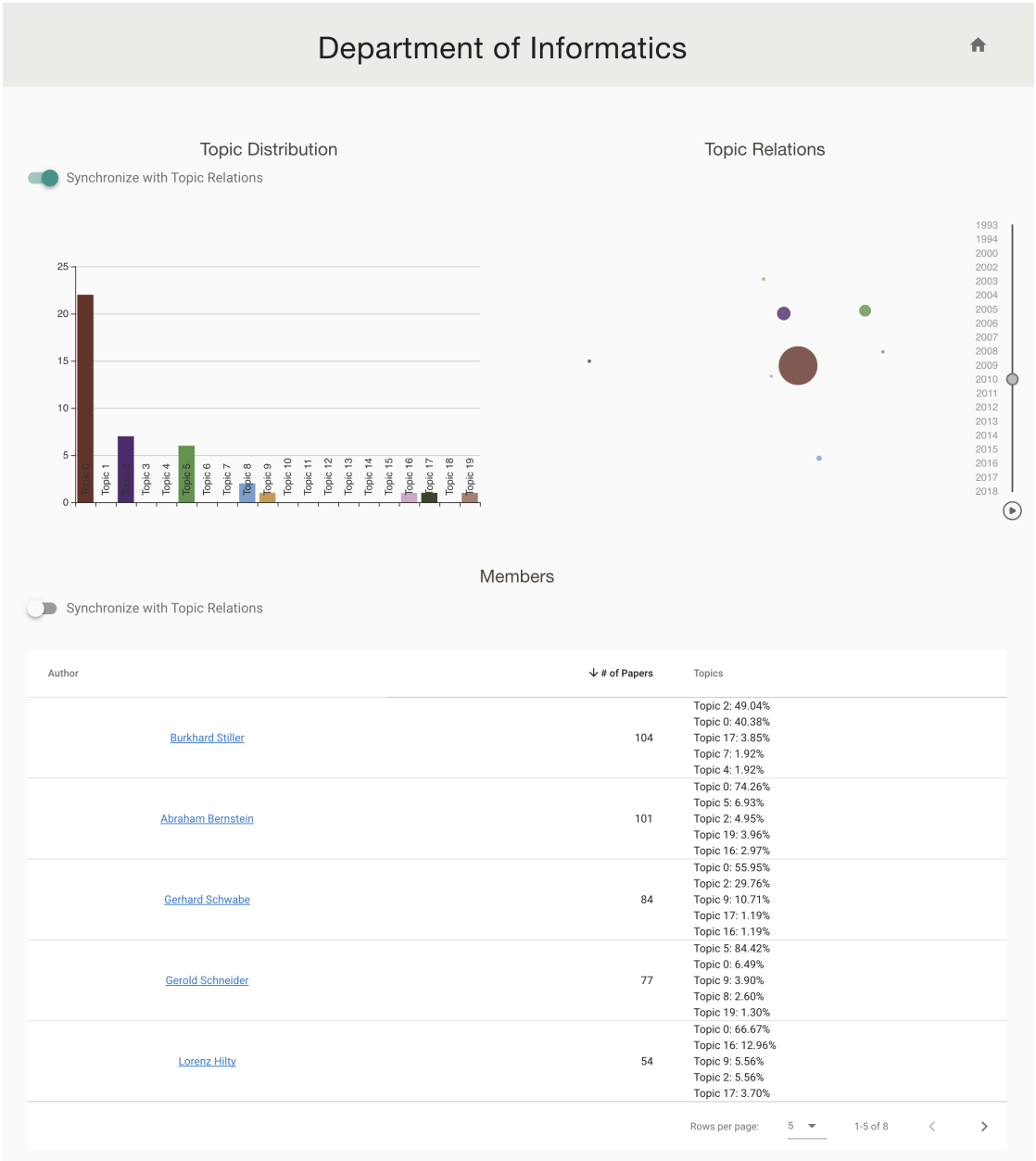


Figure 5.7: The detail page of the institute "Informatics".

The Institute Page. The institute page, shown by Figure 5.7, depicts the detail page of the Informatics department. For all the charts on this page, except the topic relation chart, the charts can either be synchronized or not. If the charts are synchronized, they will only show data related to the current year the topic relations chart is displaying at the moment. If the charts are not synchrinzed, they will use the total amount of papers for the visualisation.

On the top left, a bar chart shows the topic distribution. Each bar corresponds to a topic, which shows how many documents contributed by the institute belong to a certain topic.

On the top right, there is a scatter chart which shows the topics in relation to each other for a certain year. The bigger the topic circle is, the more documents belong to the topic.

On the bottom, there is a table which shows all the authors belonging to the institute with their contribution rate. The topics column is in relation to each author. This means that it will sum up to 100% for each of the authors.

The Faculty Page. The faculty page is identical to that of an institute, except that there is another table on the bottom left which shows the contribution per institute, the same way as the contribution per member is shown.

The Topic Page. Figure 5.8 highlights the details about the extracted topics. As explained above, only documents which belong to the topic by at least 5% are being considered relevant for a topic. Therefore, only these papers are being considered for this page. On the top left, a line chart shows the contribution trends for the topic per year. The authors can be excluded by unselecting them from the legend above. This does however only affect the line chart on the top left.

On the top right, all of the topics contributors are listed with their contribution percentages.

On the bottom left, a spider chart is shown which highlights the influence other topics have on this topic. The influence is calculated by the other topics this topics documents are part of. The intuition behind it is that a document, which contributes to topic A and topic B, topic A has to have some kind of relationship with topic B.

On the bottom right are the top ten extracted keywords, sorted by their importance. The topmost keyword is the one which defines the topic the best according to the topic model algorithm.

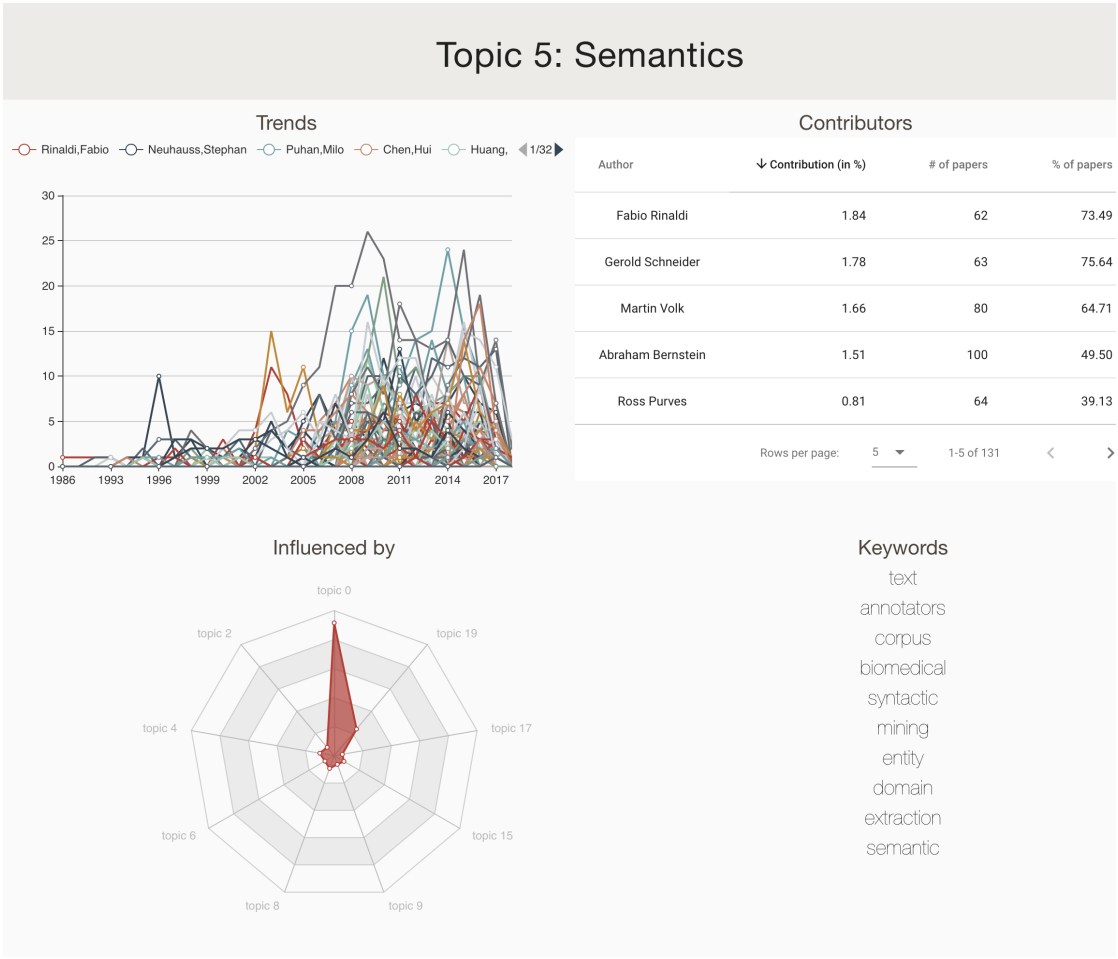


Figure 5.8: The detail page of topic 5. The topic label was chosen by hand from a list of recommendations created by the algorithm mentioned above.

6

Future Work

Labelling. While the topic labelling procedure, discussed in section 5.2, provides labels which are more interpretable than the keywords provided by the topic models, this procedure can still be improved further. One potential improvement would be to use Doc2Vec to map the documents to the nearest wikipedia articles and then choose the first common parent category of all the abstracts of one topic, as [Kataria et al., 2011] did to improve the entity disambiguation of documents. The problem is, that there are no freely available pre-trained Doc2Vec representations of the Wikipedia corpus and training one would take several days if not weeks. A simpler improvement might be to try different values for the categories the keywords are mapped onto. The most common newspaper or research categories may be potential candidates which could improve the interpretability.

Hierarchical Clusters. The visualisation could be improved by adding hierarchical clustering [Blei et al., 2003a, Kataria et al., 2011]. A possible implementation would show different topics for different zoom levels, e.g., if the user zoomed into topic 2, there would be further clustering of topic 2 into subtopics 2.1, 2.2 and 2.3 which then may provide a lot more information about the data.

Overlapping clustering for Doc2Vec. The topic model which pairs Doc2Vec with k-Means and NMF can also be improved. At the moment, the model is only able to do non-overlapping clusters. However, there are certain algorithms which can create overlapping clusters [Baadel et al., 2016], which would remove one of the major downsides of the model. Also, there might be a better way to perform the keyword extraction than to use NMF, since the data sets tend to be very small on which NMF is run on. However, in our tests, NMF performed significantly better than LDA. One idea could be to extract pre-trained Word2Vec embeddings which are in the centre of each cluster.

CNMF. The combination of CNN and NMF, which sadly couldn't convince, should not be discarded entirely, as there is many related work, as mentioned in Section 2.3, which were able to improve on the state-of-the-art models. It may be that the loss function need to be changed, however, it is important that both parts depend on each other. Tests were made where the CNN learned the matrix D, which then replaced matrix W

of the NMF part and the algorithm didn't perform well at all. Also, the algorithm is very sensitive to the learning rate and the weight and regulation constants. Perhaps, some additional values need to be tested, even though a grid search was performed for the best values.

New possible Topic Model. Another possible topic model algorithm is based on the work of [Liang et al., 2016]. What they do is to improve matrix factorisation by adding a factor which is able to capture contextual information about the data. The idea is similar to what the CNMF algorithm should have been able to do, but the approach is different. [Levy and Goldberg, 2014] were able to show, that skip-gram Word2Vec trained with negative sampling value of k is equivalent to the implicit factorising of a pointwise mutual information matrix shifted by log k:

$$\text{PMI}(i,j) = \log \frac{P(i,j)}{P(i)P(j)}$$

which can be estimated as:

$$\text{PMI}(i,j) = \log \frac{\#(i,j) \cdot D}{\#(i) \cdot \#(j)}$$

where $\#(i,j)$ is the number of times word j appears in the context of word i, $\#(i) = \sum_j \#(i,j)$, $\#(j) = \sum_i \#(i,j)$ and D is the total number of word-context pairs. Now, since this is equivalent to Word2Vec, and Doc2Vec is nothing more than an extension on Word2Vec, this formula should be adoptable to Doc2Vec, which then would mean, that the context information of a document can be approximated by a formula. This formula could then be used to improve the NMF algorithm.

7

Conclusions

This work studies different topic models on a restricted data set consisting only of abstracts of scientific publications from a multidisciplinary group of authors. In total, there were four topic models tested, LDA and NMF, being state-of-the-art topic models which are widely used and Doc2Vec + k-Means and CNMF, topic models which are new context-sensitive prototypes, explicitly developed for this work. Of the four topic models, NMF performed the best on the abstracts, followed by LDA, then Doc2Vec + k-Means while CNMF being the one performing the worst on the documents. The authors however believes, that the concept behind CNMF still is valuable and probably can be improved, maybe even to an algorithm which will outperform the others someday.

The topic model analysis is followed by a visualisation task which improves the interpretability of the extracted data. For that, t-SNE is used to map the document representations extracted from the topic model to a two-dimensional space. This resulting coordinates are displayed as a hexagon grid on a web application. Additionally, pages for detailed information of the authors as well as the institutes and faculties are also developed to improve the interpretability. To further improve the interpretability of the data, a topic labelling recommendation system is implemented which wikipedia categories are mapped to the keywords. This is done by utilising Word2Vec, which represents related words close to each other in a vector space.

Even though the new topic models, Doc2Vec + k-Means and CNMF, didn't perform well enough to replace the state-of-the-art algorithms, they still provide valuable information on how a future topic model could work as well as a solid fundament which can be improved. Also, the current topic model algorithms are good enough to capture valuable informations about abstracts (at least in some cases) as the web application shows, which is currently functional and live, and can serve the purposes of DSI for helping identify potential collaborations, and for providing an overview of ongoing research.

References

[Abadi et al., 2015] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[Anupriya and Karpagavalli, 2015] Anupriya, P. and Karpagavalli, S. (2015). Lda based topic modeling of journal abstracts. In *2015 International Conference on Advanced Computing and Communication Systems*, pages 1–5.

[Baadel et al., 2016] Baadel, S., Thabtah, F., and Lu, J. (2016). Overlapping clustering: A review. In *2016 SAI Computing Conference (SAI)*, pages 233–237.

[Blei, 2012] Blei, D. M. (2012). Probabilistic topic models. *Commun. ACM*, 55(4):77–84.

[Blei et al., 2003a] Blei, D. M., Jordan, M. I., Griffiths, T. L., and Tenenbaum, J. B. (2003a). Hierarchical topic models and the nested chinese restaurant process. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, pages 17–24, Cambridge, MA, USA. MIT Press.

[Blei et al., 2003b] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003b). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

[Boutsidis and Gallopoulos, 2008] Boutsidis, C. and Gallopoulos, E. (2008). Svd based initialization: A head start for nonnegative matrix factorization. *Pattern Recogn.*, 41(4):1350–1362.

[Boyd-Graber et al., 2017] Boyd-Graber, J., Hu, Y., and Mimno, D. (2017). Applications of topic models. 11:143–296.

[Bruggmann, 2017] Bruggmann, A. (2017). Themenlandschaften des hls. Last checked on Jan 28, 2018.

[Bruggmann et al., 2013] Bruggmann, A., Salvini, M. M., and Fabrikant, S. (2013). Cartograms of self-organizing maps to explore user-generated content. In *26th international cartographic conference*, pages 25–30.

[Bullinaria, 2004] Bullinaria, J. A. (2004). Self organizing maps: Fundamentals. *http://www.cs.bham.ac.uk/~jxb/NN/l16.pdf*. Last checked on Jan 20, 2018.

[Chen, 2011] Chen, E. (2011). *http://blog.echen.me/2011/08/22/introduction-to-latent-dirichlet-allocation/*. Last checked on Jan 23, 2018.

[Colyer, 2016] Colyer, A. (2016). The amazing power of word vectors. Last checked on Jan 23, 2018.

[Commons, 2008] Commons, W. (2008). Latent dirichelt allocation. Last checked on Jan 23, 2018.

[Hughes et al., 2017] Hughes, M., Li, I., Kotoulas, S., and Suzumura, T. (2017). Medical text classification using convolutional neural networks. *CoRR*, abs/1704.06841.

[Karn, 2016] Karn, U. (2016). *https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/*. Last checked on Jan 22, 2018.

[Kataria et al., 2011] Kataria, S. S., Kumar, K. S., Rastogi, R. R., Sen, P., and Sengamedu, S. H. (2011). Entity disambiguation with hierarchical topic models. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 1037–1045, New York, NY, USA. ACM.

[Kim et al., 2016] Kim, D., Park, C., Oh, J., Lee, S., and Yu, H. (2016). Convolutional matrix factorization for document context-aware recommendation. pages 233–240.

[Kim, 2014] Kim, Y. (2014). Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

[Kohonen, 1990] Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480.

[Koltcov et al., 2014] Koltcov, S., Koltsova, O., and Nikolenko, S. (2014). Latent dirichlet allocation: Stability and applications to studies of user-generated content. In *Proceedings of the 2014 ACM Conference on Web Science*, WebSci '14, pages 161–165, New York, NY, USA. ACM.

[Le and Mikolov, 2014] Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. *CoRR*, abs/1405.4053.

[Lee and Sebastian Seung, 1999] Lee, D. and Sebastian Seung, H. (1999). Learning the parts of objects by non-negative matrix factorization. 401:788–91.

[Lee and Dernoncourt, 2016] Lee, J. Y. and Dernoncourt, F. (2016). Sequential short-text classification with recurrent and convolutional neural networks. *CoRR*, abs/1603.03827.

[Levy and Goldberg, 2014] Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2177–2185, Cambridge, MA, USA. MIT Press.

[Liang et al., 2016] Liang, D., Altosaar, J., Charlin, L., and Blei, D. M. (2016). Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 59–66, New York, NY, USA. ACM.

[Lopez and Kalita, 2017] Lopez, M. M. and Kalita, J. (2017). Deep learning applied to NLP. *CoRR*, abs/1703.03091.

[M Blei, 2011] M Blei, D. (2011). Introduction to probabilistic topic models. 55.

[Maaten and Hinton, 2008] Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

[Mikolov et al., 2013a] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

[Mikolov et al., 2013b] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.

[Newman et al., 2010] Newman, D., Lau, J. H., Grieser, K., and 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*, HLT '10, pages 100–108, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

[Pinto Avendano et al., 2006] Pinto Avendano, D., Jimenez, H., and Rosso, P. (2006). Clustering abstracts of scientific texts using the transition point technique. pages 536–546.

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

[SKLearn, 2017a] SKLearn (2017a). *http://scikit-learn.org/stable/modules/decomposition.html#latentdirichletallocation*. Last checked on Jan 21, 2018.

[SKLearn, 2017b] SKLearn (2017b). *http://scikit-learn.org/stable/modules/decomposition.html#nmf*. Last checked on Jan 21, 2018.

[TF-IDF, 2017] TF-IDF (2017). *http://www.tfidf.com/*. Last checked on Jan 22, 2018.

[Underwood, 2014] Underwood, T. (2014). *https://tedunderwood.com/2012/04/07/topic-modeling-made-just-simple-enough https://web.archive.org/web/20180102164400/https://tedunderwood.com/2012/04/07/topic-modeling-made-just-simple-enough/*. Last checked on Jan 20, 2018.

[Weiss et al., 2004] Weiss, S., Indurkhya, N., Zhang, T., and Damerau, F. (2004). *Text Mining: Predictive Methods for Analyzing Unstructured Information*, chapter Case Studies. SpringerVerlag.

[Yan et al., 2013] Yan, X., Guo, J., Lan, Y., and Cheng, X. (2013). A biterm topic model for short texts. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1445–1456, New York, NY, USA. ACM.

A

Appendix

A.1 Example of a JSON export from ZORA

```
{
  "eprintid": 87751,
  "date": 2013,
  "documents": [{
    "uri": "http://www.zora.uzh.ch/id/document/359737",
    "format": "application/pdf"
  }],
  "creators_abbrv": [
        {
          "lineage": null,
          "given": "G",
          "honourific": null,
          "family": "Schneider"
        }
  ],
  "language_mult": [
    "eng"
  ],
  "abstract": "The investigation of specific...",
  "type": "article",
  "title": "Describing Irish English with the ICE Ireland
     Corpus"
}
```

Listing A.1: JSON export of a single document from ZORA

Listing A.1 shows an example of a single document as JSON, exported from ZORA. The JSON contains a lot of information and this listing is therefore shortened to only represent the information of interest for this work. The whole file can be found on the CD accompanying the work.

- eprintid: Each document should have an id which uniquely identifies the document in the database. Since ZORA already provides such an id, this id is taken over to the database of this work. This also allows the web application to link a document to the corresponding site on ZORA.
- date: The date when the work was published. This can be empty or in various different formats, like "2013" or "August, 2013". Therefore, only the year, if available is parsed by a regular expression.
- documents: The documents array can contain multiple different documents, each of which has a format description as well as a url, where the document can be accessed. However, the access rights may be restricted. Also, the documents array might contain noise (e.g, a coversheet which is a pdf).
- creators_abbrv: The abbreviations of all the authors of a document. However, this abbreviations seem to be user entered, as they can differ from document to document.
- language_mult: This identifies the language of the main document. However, the abstracts may be in another language or even in multiple languages.
- abstract: The abstract of a document, if available
- type: The type of the document.
- title: The title of the document.

B

Appendix

B.1 CD Content

The accompanying CD contains the following:

- code/: The source code of the project
- Related work/: A collection of related work referred in this project
- Paper/: Contains two text files "Zusfsg" and "Abstract" as well as a complete pdf document of this thesis called "Bachelorarbeit"
- Presentation/: A presentation about the project

B.2 Installation Guide

The following dependencies are required to run the application:

- Anaconda[1]
- NodeJS[2]

Using a UNIX terminal, navigate to the content of the CD by using the following command:

```
$ cd code
```

The code-folder contains the following subfolders: webapp and topic extraction. To run the web application, enter the following commands:

```
$ cd webapp/backend
$ npm install
$ npm start
```

[1] https://anaconda.org/anaconda/python, last accessed on Jan 29, 2018
[2] https://nodejs.org/en/, last accessed on Jan 29, 2018

This will navigate you to the subfolder where the backend is located, updates the node dependencies and starts the backend. Then, in a new UNIX terminal, enter the following commands:

```
$ cd code/webapp/client
$ npm install
$ npm run dev
```

This will navigate you to the subfolder where the client is located, updates the node dependencies and starts the client in developer mode. The page will then open inside your browser. To build the client for production, the following command can be run inside the client's folder:

```
$ npm run build
```

This will create the files for the web application in the dist folder. The files needed to be uploaded to a server/web space to display the web application are located at:

- client/index.html
- client/dist/*
- client/node_modules/vuetify/dist/vuetify.min.css
- client/node_modules/tippy.js/dist/tippy.css

To manually run the topic extractor, enter the following into a UNIX terminal:

```
$ cd code/topic_extractor
$ conda create --name topic_extractor
$ source activate topic_extractor
$ python
$ >>> import nltk
$ >>> nltk.download('stopwords')
$ >>> quit()
$ python main.py
```

This will create a anaconda environment, download and install all the python dependencies for the environment, download the used stopwords by nltk and then starts the topic extraction process.

List of Figures

2.1 The LDA topic model, represented as a plate notation. From [Commons, 2008] 8
2.2 Concept of NMF for Topic Modelling 9
2.3 Concept of t-SNE 13

3.1 Language Distribution on Research Material of Interest 15
3.2 Quality of the Data 16
3.3 Document Distribution among faculties 17
3.4 Document Distribution among institutes 18
3.5 Document Distribution among authors 18
3.6 Most common words across the documents 20

4.1 Overview of CNMF: A Model which enhances NMF through a CNN . . . 35
4.2 Result from Word Embedding Layer 36
4.3 Convolutional Layer 37

5.1 The paper distribution created by t-SNE, labelled after the topic they contribute the most. 44
5.2 The main page. The topic map shows a global overview about the topics. The green dots are some of the authors which contributed the most documents. 46
5.3 The main page zoomed in to reveal more of the structure. All of the authors (green dots) as well as the papers (red dots) are shown 47
5.4 The main page focused on the Institute for Computer Linguistics. The tiles where a lot of papers are placed are red, the ones with only a few or none papers are blue. Also, only the contributors belonging to the institute are shown. 49
5.5 The detail page of author Gerold Schneider. The page consists of interactive charts which make the data specific to the author more interpretable. 50
5.6 The detail page of an author, filtered by the year 2012. 51
5.7 The detail page of the institute "Informatics". 52
5.8 The detail page of topic 5. The topic label was chosen by hand from a list of recommendations created by the algorithm mentioned above. . . . 54

List of Tables

4.1 Top 5 Keywords per Topic for LDA 25
4.2 Top Contributors per Topic for LDA 26
4.3 Top 5 Keywords per Topic for NMF 28
4.4 Top Contributors per Topic for NMF 29
4.5 Top 5 Keywords per Topic for Doc2Vec 32
4.6 Top Contributors per Topic for Doc2Vec 33

ärung Due to the fast increasement of available documents in the Internet, methods are needed which are able to present the content of the data, without the need to read them. This methods already exists, called topic models, but tend to work only for large documents. This work analyses current state-of-the-art topic models as well as presenting some own, context-sensitive approaches on a restricted data set built from abstracts. Then, the best results are visualised to improve the interpretability of the data.