Freie Universität Berlin

Bachelorarbeit am Institut für Informatik der Freien Universität Berlin

Human-Centered Computing (HCC)

Comparing interpretability techniques for unsupervised topic modeling

Tim Korjakow

Matrikelnummer: 372862

Email: tim.korjakow@campus.tu-berlin.de

Betreuer: Jesse Jonas Benjamin

Betreuerin und Erstgutachterin: Prof. Dr. C. Müller-Birn

Zweitgutachter: Prof. Dr. K. Müller

Berlin, 08.08.2019

b

Eidesstattliche Erklärung

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Berlin, den July 29, 2019

<Name>

Abstract

<Please summarize your thesis in a brief but meaningful way (about one page). Include in your abstract the topic of this thesis, important contents, results of your research and an evaluation of your results.>

Zusammenfassung

<Hier sollten Sie eine kurze, aussagekräftige Zusammenfassung (ca. eine Seite) Ihrer Arbeit geben, welche das Thema der Arbeit, die wichtigsten Inhalte, die Arbeitsergebnisse und die Bewertung der Ergebnisse umfasst.>

Contents

List of Figures

List of Tables

Vorwort

Allgemeine Hinweise zur Erstellung einer Abschlussarbeit

- Beachten Sie, dass diese Vorlage für einen zweiseitigen Ausdruck angelegt wurde.
- Über die Papierqualität können Sie entscheiden, aber wir empfehlen aber Seiten mit wichtigen, farbigen Grafiken auch in Farbe auszudrucken und dabei ein höherwertiges Papier zu verwenden.
- Bitte stimmen Sie mit dem Betreuer Ihrer Arbeit auch den Zweitgutachter ab. Die Anfrage des Zweitgutachters erfolgt von Ihnen. Es ist an dieser Stelle sinnvoll, die Anfrage mit einer kurzen Zusammenfassung der Arbeit zu stellen.
- Bitte beachten Sie, dass Sie Ihre Abschlussarbeit mit einer Klebebindung versehen, eine Ringbindung ist nicht erwünscht.

1 Introduction

1.1 Project IKON

This thesis has a direct application in a project which tries to explore potentials for knowledge transfer activities at a research museum. Project *IKON* was started in cooperation with the German Natural History Museum in Berlin which houses more than 600 [**TK:** Right number?] scientists, PhD students and other staff. With that size of scientific staff the institution is a global player in research on evolution and biodiversity [Int]. Despite its importance in the research landscape, the museum is challenged with a lack of shared knowledge across working groups and organizational structures such as departments. In interviews researchers from the project were able to trace these problems back to the very intricate and complex layout of rooms and halls in the building which was originally constructed in 1810. In order to mitigate this problem Figure 1.1 shows one of the main deliverables of *IKON* - a ML-driven data visualization which follows the path of knowledge at this research museum from its creation in projects over knowledge transfer activities, where multiple projects exchange their findings and try to generate added value for each other, to the final target group. Knowledge transfer is made explicitly visible by showing projects not in the predefined taxonomy of the museum, but instead in semantic relation to each other. This is accomplished by running all project abstracts through a topic modeling process consisting of four major components, as seen in Figure 1.2.

1.1.1 Topic modeling

A general topic modeling pipeline consists of four steps:

1. Document embedding
2. Topic extraction
3. Classification of documents
4. Reduction into 2D

Given an unlabeled corpus $C = \{D_1, ..., D_n\}$ consisting of n documents $D_i = (t_1, ..., t_m)$, which in turn consists of a sequence of m strings, also called tokens or words, the document embedding step assigns to each document a vector $v_D \in \mathbb{R}^e, e \in \mathbb{N}^+$. Semantically similar documents should also be closer in the embedded vector space with respect to a given distance measure than

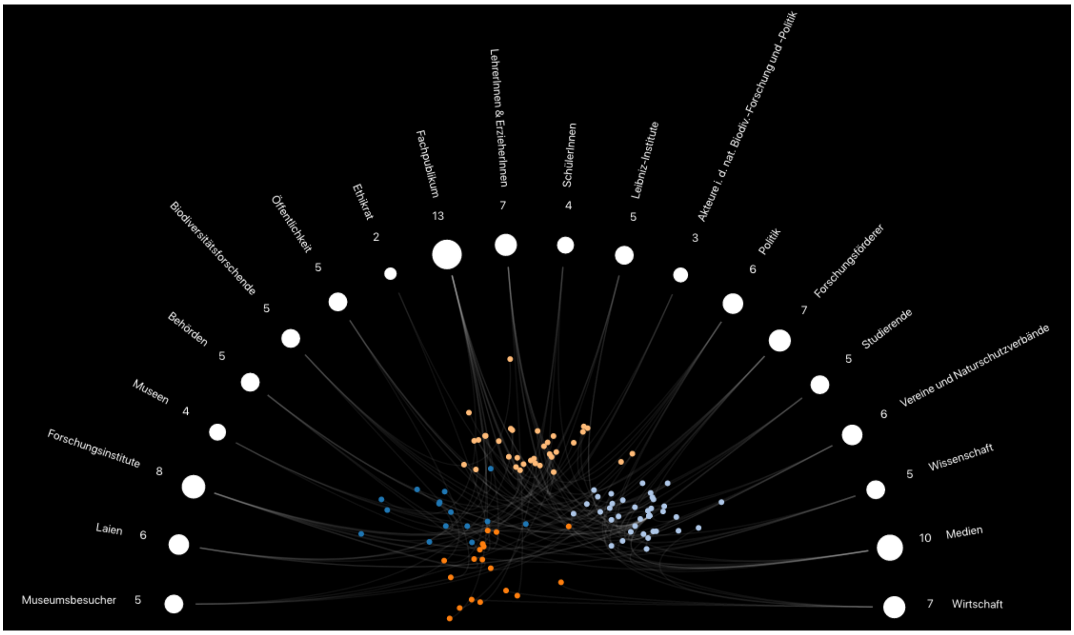

Figure 1.1: Screenshot of the cluster view of the IKON visualization

documents which are semantically not related. Therefore this step transforms a corpus into a matrix $(v_1, ..., v_n) \in \mathbb{R}^{e \times n}$.

Consuming the output from the previous step the topic extraction tries to uncover k latent structures. We call these structures *topics*. Mathematically speaking a topic is a probability distribution over a fixed set of input features. [LTD+16] These features can correspond to tokens, as it is the case in the later discussed Tfidf-BOW embedding, but this does not have to be the case. Therefore this step transforms the corpus from the embedding space of dimensionality $e \times n$, where each document is described as linear combination of features, to the latent space of dimensionality $k \times n$, where each document is described as a linear combination of latent topics. Since most often $k < e$ holds true, this can also be seen as a form of dimensionality reduction, which is again a form of feature extraction.

Using the document vectors in the latent space each document is assigned a label. This may happen in a supervised way if there are labels available for training purposes, but in most cases an unsupervised classification, also known as clustering, is used to group the documents.

Finally in order to visualize the high dimensional distribution of documents in the latent space another dimensionality reduction is used to project the documents to 2D.

First user tests and interviews unveiled that, even though the visualization was specifically tailored to non-technical users [**TK:** neeeds definition], the scientists from the museum had a hard time interpreting and understanding the output generated by the pipeline. Furthermore each component in Figure 1.2 introduces additional parameters which influence the results generated by the

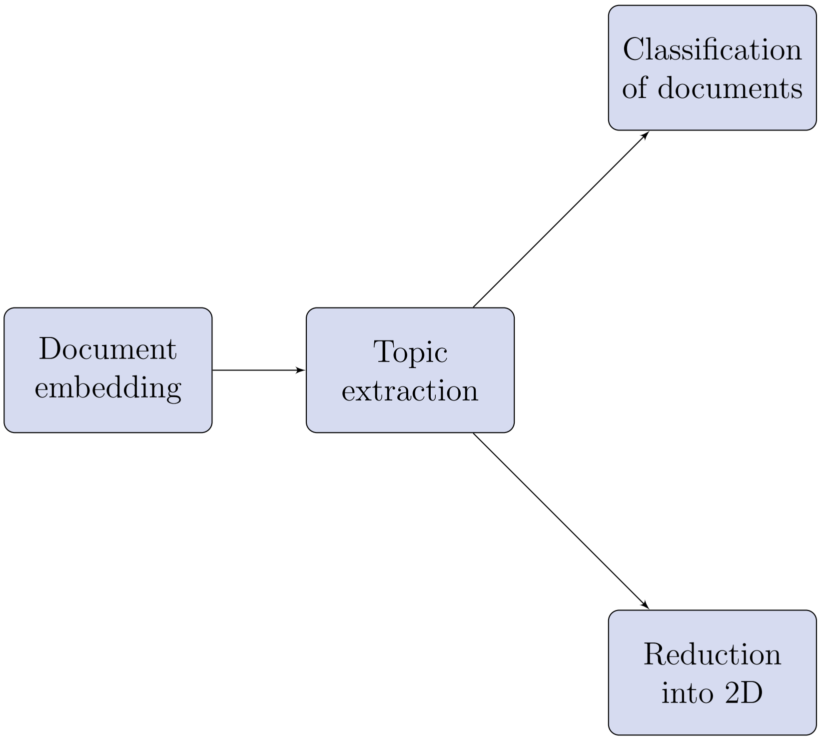

Figure 1.2: Components of a general topic extraction pipeline

pipeline.

In order to lay the groundwork for this thesis and understand the challenges which scientists face while interacting with the visualization I carried out a workshop with the researchers from project *IKON*. In the beginning I asked them which kind of hardships they, based on their past experiences and interviews, observed during the interaction between user and visualization. Followed by an explanation of Figure 1.2 we discussed how these challenges may correlate with goals and questions. Following a description of the key questions each question was categorized according to the pipeline step, , as seen in Table 1.1, which may contribute information in order to support the user in answering his question.

1.2 Interpretability

With the surge of the application of machine learning (ML) systems in our daily life there is an increasing demand to make operation and results of these systems interpretable for people with different backgrounds (ML experts, non-technical experts etc.). Contrary to these efforts, interpretability as term has become an ill-defined objective [Lip16] for research and development in ML algorithms since there is no widely agreed upon definition of it. This leads to a very fragmented nature of the field.

Miller et al. [MHS17] support this point by conducting a literature study

Question	Applicable pipeline component
How does the research landscape look like and on what kind of topics are prominent?	Topic Extraction
What does a cluster mean?	Classification
What does the distance between clusters/projects mean?	Topic Extraction / Reduction into 2D
How similar are two projects/clusters?	Topic Extraction

Table 1.1: Table showing the sourced questions and the pipeline step which could provide an answer

and uncovering that interpretability research is rarely influenced by insights from the humanities, especially connected fields as explainability or causality research.

[**TK:** Interpretation of machine learning (ML) results is a major challenge for humans, especially for non-technical experts [ref]. Research on interpretability[1] in the ML community has focused on developing interpretability techniques, i.e. specific technical approaches to generate explanations[2] for ML results. However, applications of these techniques are predominantly concerned with making particular model features understandable, rather than supporting the interpretation of ML-driven systems in a specific context of use. At the same time, research in the HCI domain often remains on a formal, algorithmic level—explanations tend to be technical and tailored to an expert audience, mirroring the technical focus of ML research. Realistic use cases and qualitative, context-aware evaluations to inform the selection and design of interpretability techniques remain rare. While we do not see complete transparency as a prerequisite for interpretability we hypothesize that in general, since interpretation is dependent on context, interpretability techniques cannot be fully context agnostic either. Therefore, our general approach is to research interpretability from a context-aware perspective, i.e. we explore how interpretability can be operationalized in a specified, well-defined domain context.]

[1] Which we position to be a high-level precondition for Explainability from the XAI [**?**] and Fairness, Accountability and Transparency, from the FAT-ML discourse [**?**].

[2] Which we define as instances of interpretability techniques.

2 Literature mapping study

2.1 Motivation

In order to access current methods in the fast-moving field of interpretability research in machine learning in a reproducible and structured fashion I will conduct a literature mapping study according to Petersen et. al [PFMM], which consists of a number of sequential steps which should result in a representative corpus and an analysis using it.

2.2 Methodology

The recommended process is augmented by further steps in order to tailor it to the existing use case and consists of the following seven procedures:

1. Definition of research questions:

   The overall process starts by defining clear questions which should guide the development of the whole literature mapping study and subsequently the result as well. Since I am interested in gaining an overview over the existing interpretability techniques, I chose the following questions:

   a) What categories of explainability techniques are mentioned in the corpus?

   b) What kind of models are enhanced by explainability techniques?

   c) Which techniques are applicable to results produced by the pipeline or the pipeline itself?

2. Construction of a search string:

   Based on the questions one is able to gather a set of key words which are most relevant to the field which is analyzed. Each word is augmented by synonyms which are concatenated with boolean OR operators and several of these synonymous groups are again connected via logical ANDs. Applying this method to the previously found questions yields the following search string:

   ("explainability" OR "explainable" OR "explanation" OR "explaining" OR "interpretability" OR "interpretable" OR "interpretation" OR "interpret" OR "understanding") AND ("machine learning" OR "neural network" OR "neural networks" OR "AI" OR "XAI" OR "artificial intelligence" OR "model") AND ("text" OR "document" OR "NLP" OR "natural language programming" OR "review" OR "method" OR "technique" OR "visualization")

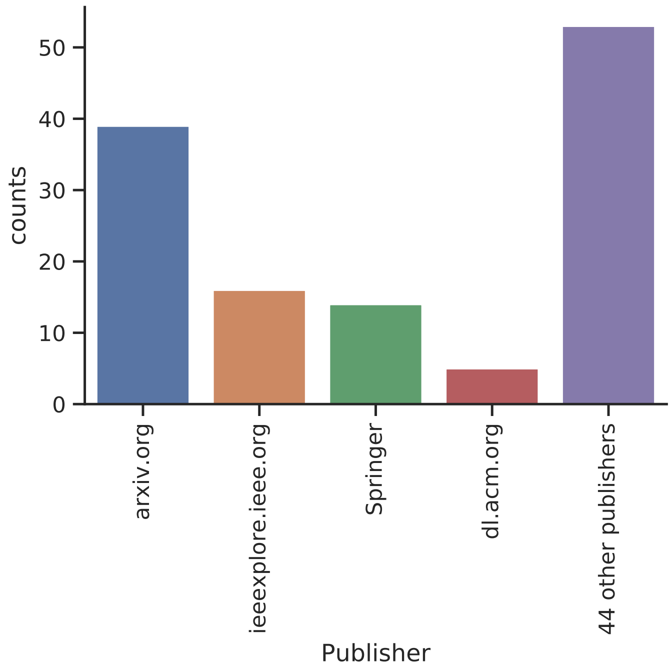

Figure 2.1: Barplot displaying the distribution of publishers occurring in the meta search results

3. Analysis of the main publishers using a meta search and the search string:

   Due to the presumed distributed nature of interpretability research it is not easy to pinpoint the main publishers of scientific articles. In order to mitigate this, a pre-search in the meta-search engine 'Google Scholar' is conducted. It should be noted at this point that any biases which are apparent in the meta search engine therefore apply to this analysis as well. One can see in Figure 2.1 that the main publishers are respectivly Arxiv, IEEE, Springer and ACM. Since all of these publishers are mainly focused on publications in computer science, mathematics and engineering, this speaks in favor of the hypothesis that most of the research is still very technical and research from social sciences rarely influences it. Even though Arxiv is not a credible publisher per se, it seems like the research community uses it as the first place to publish work and therefore it should not be excluded in this analysis.

4. Sourcing of publications in scientific databases:

   Based on the insights from the previous step each of the main publisher's databases is scraped using the search string and their respective 'advanced search' interfaces or their APIs. Since most searches result in more than 1000 publications only the top 100 results ordered by the relevance scoring of the database are taken into account. These publications then form the corpus which is the basis for further analysis.

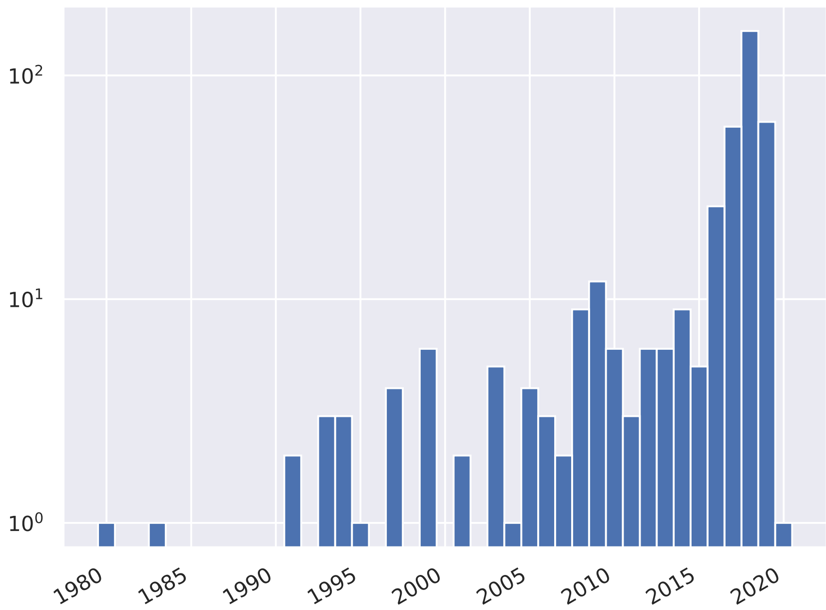

Figure 2.2: Barplot displaying the distribution of publishers occurring in the meta search results

5. Intermediate assessment of the corpus:

   Looking at the distribution of tags in Figure 2.3 it is apparent that the chosen keywords represent the field well. There are no tags in the first 5 entries which are not constructable by the query. Plotting the distribution of publishing dates of the papers from the corpus in Figure 2.2 reveals that the first publications were already written in 1980, while there is a surge of interest and research in the last 4 years. This speaks in favor of the premise that interpretability research is not necessarily a young, but a recently thriving field.

6. Definition and application of inclusion and exclusion criteria to narrow down the pool of publications further:

   The next step serves as another filtering step enhancing the quality of the hitherto automatic selection by using human decision making. A combination of the guiding questions, which were defined in the beginning of the process and a first pass over the whole corpus, in which I skimmed the papers, gave me a clear set of criteria, as seen in Table 2.1, which can be used to filter the corpus further. In a second pass each paper was evaluated and included in the next step if and only if it satisfied at least one inclusion criterion and none of the exclusion criteria. In order to support my decision making and minimize the amount of work to classify each paper I developed a Jupyter-based interface, which takes a bibliography and a set of inclusion and exclusion criteria and iterates over all contained publications, shows its title and abstract and allows the user

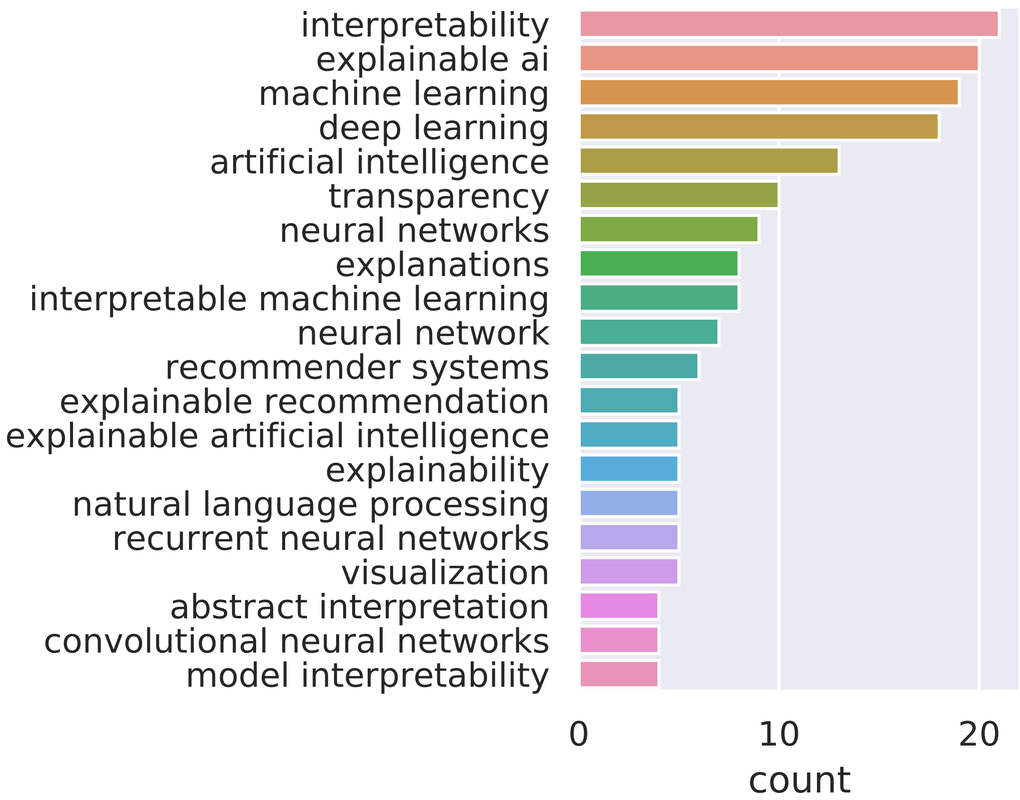

Figure 2.3: List of the 20 most used tags and their absolute frequency

Inclusion criteria	Exclusion criteria
• Reviews the current state of explainability research • Presents a specific method for enhancing explainability for models	• Is not scientific literature • Does not describe the used explainability method • The publication does not focus on explainability • The described method is neither general, nor focused on NLP

Table 2.1: Table showing all used inclusion and exclusion criteria

to select criteria which apply. If a closer examination is needed it opens the paper on demand. Furthermore it sorts each publication into either a bibliography for the next stage, a bibliography with rejected publications depending on the applying criteria. [**TK:** Interesting? Should I show the interface?]

7. Quantitative assessment of the resulting corpus:

   In the last step the actual mapping is generated. In another pass I first skimmed and then read each paper and based on that classified each publication and its presented technique in order to answer the initially posed questions. To answer the first question I categorized them according to the proposed categories of Hohman et. al. [**?**]. These categories are not a perfect fit for a thesis dealing with explainability for non-technical experts since it also categorizes techniques according to their mathematical inner workings, but Hohman et al. extended the categories proposed by Lipton [Lip16], which formulated the starting hypothesis for this thesis and is the closest to a nontechnical assessment of interpretability research I could find. Furthermore each publication was assigned the type of model to which the technique is applicable, the component to which the technique could be applied in the topic extraction pipeline and each paper was classified as either ”Theory”, ”Method”, ”Study” or ”Report”. A ”Method” paper presents a single explainability technique and demonstrates its impact in an exemplary use case. A ”Theory” paper does so as well, but misses a presented application and evaluation. A ”Report” on the other hand summarizes and presents multiple techniques. Finally, a ”Study” paper shows the results of an interface evaluation which visu-

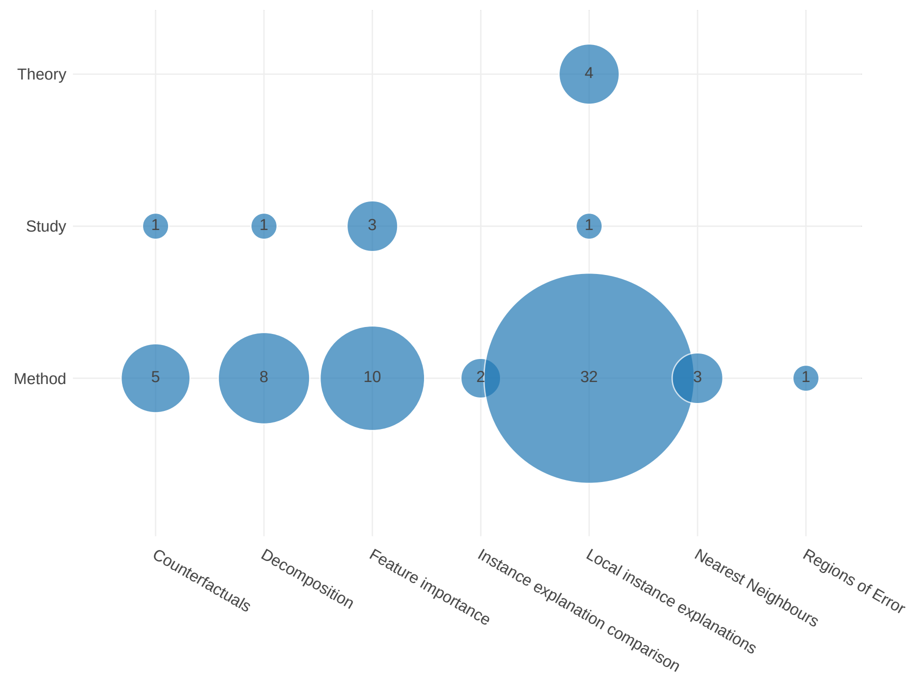

Figure 2.4: Mapping of the type of publication and its Gamuth classification

alizes the output of explainability methods. Publications from the last category are therefore less technical and more concerned with the HCI aspects of explainability techniques and their visualization.

Since most of the overview papers presented a huge amount of techniques which were already covered by the "Method" papers and the corpus was already large, I decided to exclude them from the last mapping step. This reduced the final corpus to a size of 72 publications.

2.3 Results

In order to answer my first question concerning the different kinds of researched explainability

Mapping the type of paper and the classification according to Gamuth each on an axis (Figure 2.4) shows clearly that there is a trend towards developing methods which explain single decision instances (38 paper). Furthermore most developed methods are tested on real world data (61 paper), but their application in an interface is rarely studied (6 paper). This speaks in favor of the hypothesis that most explainability methods are developed as mathematical theories and influences from HCI are rarely taken into consideration.

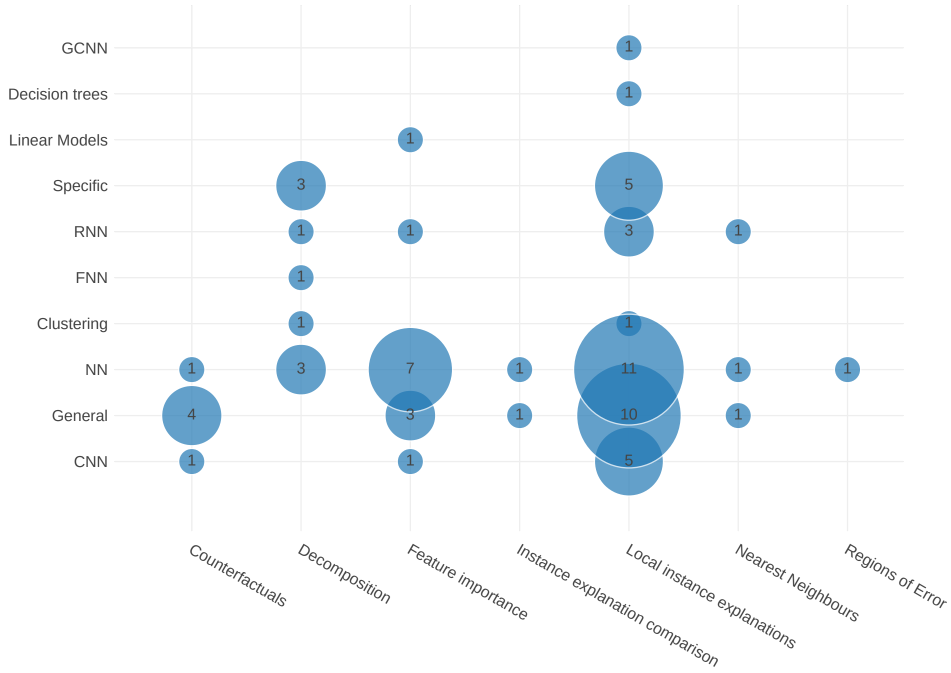

Figure 2.5: Mapping of applicability and Gamuth classification

The second question was concerned with the type of models which are enhanced by explainability techniques. In Figure 2.5 it is visible that neural architectures (NN, CNN, FNN, RNN, GCNN) dominate the field (40 paper). 19 papers try to explain a given model in an agnostic way as a black box, while a minority of publications deals with the explainability of clustering results, decision trees or linear models.

The third mapping in Figure 2.6 shows the relation between the applicability of a method in the general topic extraction pipeline and its Gamuth classification. Suprisingly, 51% of the sourced publications are not applicable to the general topic extraction pipeline in any form. The two main reasons why a publication falls into this category is that it either presents a method in a subdomain of NLP which is not directly applicable [GMPB16] [IST+18] or its presented use case and context is too far off in order to be applied [MWM18] [GCJC]. [**TK:** Is that true?] The second biggest category consists of techniques which could be applied to the document classification step using labeled data to train a model. Since any neural network can be used to classify vectorized documents, most of the publications on the "NN" axis in **??** fall into this bucket as well. All in all, 18 publications remain which could be applied to an unsupervised topic extraction pipeline. [**TK:** Explain why there is Dimreduction instead of Topic Extraction and 2D] The document embed-

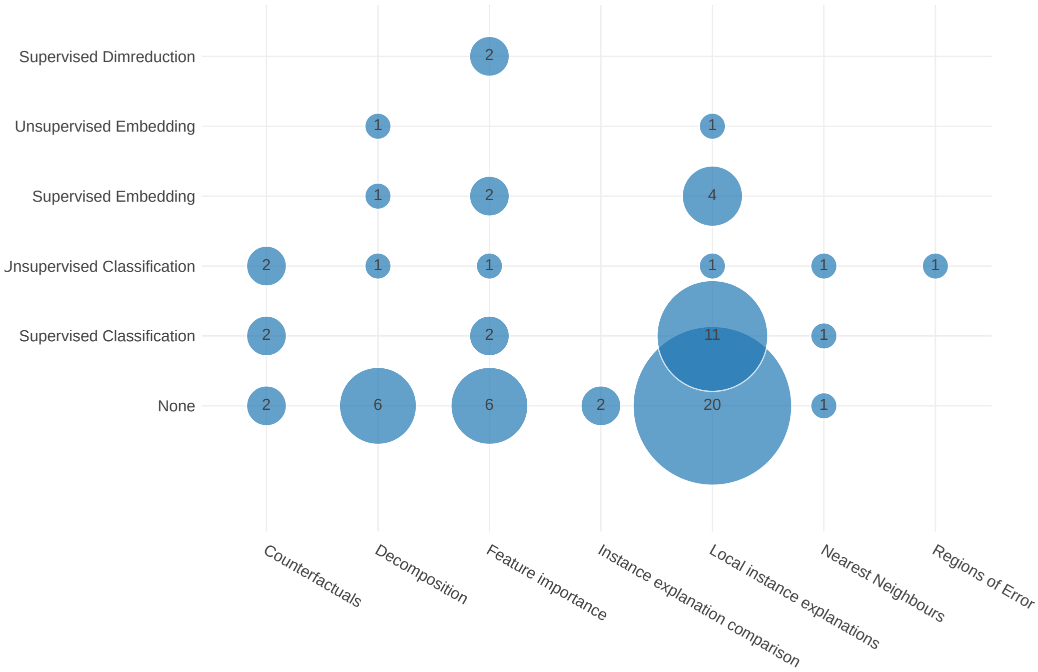

Figure 2.6: Mapping of pipeline step and Gamuth classification

ding step could be made interpretable by decomposition, feature importance visualization or by explaining the embedding of single instances.

Kim et al. [KLHK19] decompose a pretrained network and extract simple features which they use as to train a neural network on another task in a transfer learning fashion. The predictions for new tasks can then be described as a combination of these extracted features.

In contrast to that, Zhang et al. [ZYL$^+$18] train another neural network to explain the output of any given neural network in a unsupervised way. They focus on CNNs and utilize the fact that these convolutional layers contain structural information. For each input they are able to disentagle the information from the applied convolutional filters and extract features which can be applied back to the input as masks to show influential parts. Given a document embedding technique, which uses CNNs, and a corpus this explainability technique could be used to highlight influential parts of the input document.

3 Implementation

3.1 Data and Preprocessing

In order to connect projects semantically instead of by the rigid taxonomy of the museum, I was able to use the project's self-description which is recorded in the GEPRIS database of the DFG [DFG]. It consists of almost all projects which were supported by the DFG since 2000. Fortunately, another bachelor project before me worked on a scraper which extracted approximately 114.000 projects from the web interface of the database since there is no publicly available API. Each project was characterized by a title, a project abstract in German or English, start and end dates as well as additional meta data like connected institutions or people working in the project.

As one can see in Figure 3.1, there is a peak at word count 3 and one at approximatly 100. The first one corresponds to all projects which do not have descriptions, because they are described with "Keine Zusammenfassung vorhanden". The latter peak on the other hand is produced by projects from a fund which uses the same descriptions for all its projects which are financed through the DFG.

Removing these peaks in Figure 3.2 reveals that most texts have an length of 150 words, while also having smaller peaks at ca. 70 and 350 words. The shortest description has a length of one word and the longest 983 words.

Following the advice of Matthew et al. [Den17] the texts were preprocessed by a P-N-S-W scheme. First punctuation (P) and numbers (N) were removed since sentence boundaries or specific numbers do not bear a lot of information in middle-sized descriptive texts. Following this, according to the categories of Matthew et al., a stemming step (S) is performed, which uses lemmatization to find the lemmas of words by using vocabularies and the context of each word. The last step removes infrequent words without much semantic meaning, commonly known as stopwords (W). Lowercasing and n-gram inclusion were omitted, because casing is an important feature for distinguishing nouns from other word types in the German language, which helps the lemmatization step, and the use of word composition makes most reasonable n-grams in other languages appear as one word in German.

Until the start of this thesis the pipeline did all this preprocessing using regex-based rules and a lemmatization using the SpaCy lemmatizer. This proved to be a viable option until a corpus size of 5000 since after that point the running time was too long to effectively work with it. Therefore I bundled all the preprocessing operations in a new class called *Datapreprocessor*, which should be able to transform any given query into a preprocessed dataset for the following pipeline steps as well as cache its results. In order to do that I rewrote

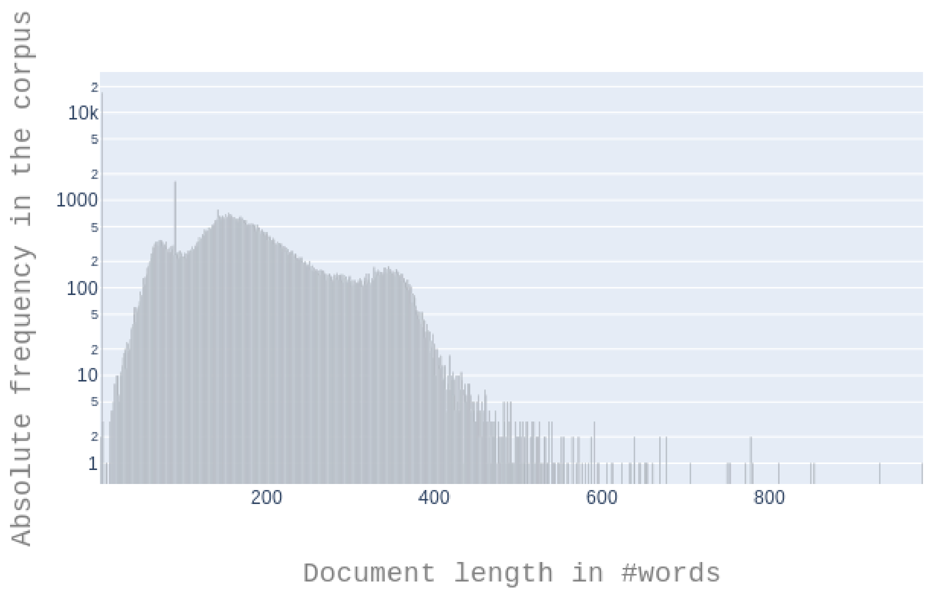

Figure 3.1: Histogram showing the distribution of text lengths in the dataset

the preprocessing steps and integrated them into the already existing SpaCy pipeline which uses a CNN to apply the previously discussed preprocessing. Additionally it is able to detect the language of a text, which, in turn, makes it possible to filter out all non-German texts. Using this existing framework gave me the opportunity to embed my custom code into the Cython code of the framework accelerating the looping over the corpus. Additionally I was able to fully parallelize the process on n CPUs by splitting the corpus in n chunks and feeding each chunk into a separate sub-process to make use of the batch sizes of the SpaCy neural networks. This accelerated the preprocessing by a factor of 10.

3.2 The existing pipeline

The existing pipeline was implemented by me as a proof-of-concept for project *IKON*. Following the structure of Figure 1.2 the first step is a document vectorization of the given texts in order to embed them in one common vector space. One of the simplest and still effective methods is a Tf-Idf Bag-Of-Words (TfIdf-BOW) embedding. With this procedure each text is represented as a set of terms, the bag of words. Having a whole corpus it is now possible to assign a vector to each document D in corpus $C = \{D_1, ..., D_n\}$ of length $N = |C|$, where each entry i is the number of term occurrences of term t_i in D. That means that each document gets embedded into a vector space of dimensionality |(unique terms in C)| and the corpus becomes a matrix of size

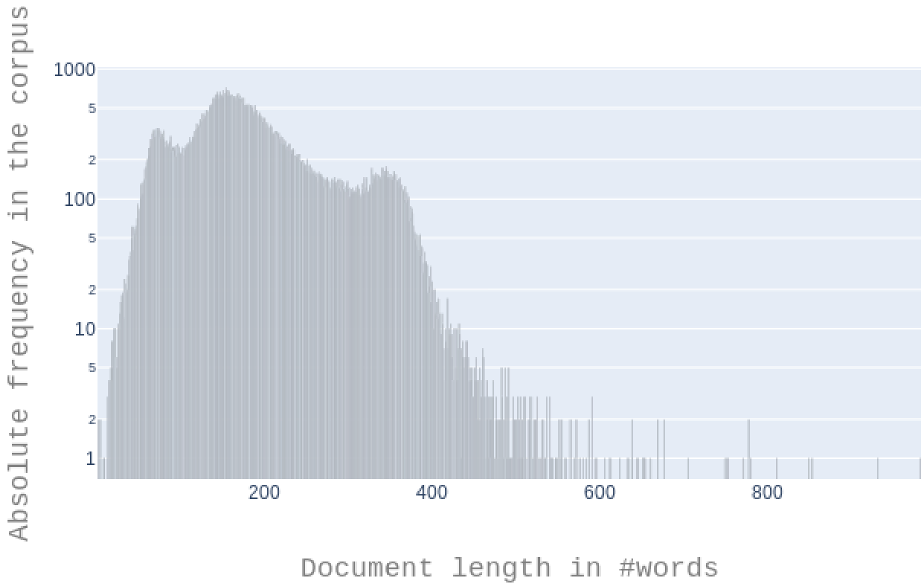

Figure 3.2: Histogram showing the distribution of text lengths in the dataset excluding duplicates and projects without a description

$|(\text{unique terms in C})| \times N$. In order to additionally introduce information from the whole corpus into each vectorized document and therefore contextualize it, each entry is replaced by $C_{t,d} = Tf(C_{t,d}) \cdot Idf(C,t,d)$ where $Tf(t,d)$ is often the identity function and $Idf(C,t,d)$ is $\log \frac{N}{|\{D \in C : t_t \in D\}|}$. [Piv] The notion behind this is intuitive. The higher the term frequency of a term in a document, the more important it is for this specific document and the more a term appears in several documents, the less it caries information to seperate a document from others. [**TK:** Needs maybe rework based on Shannon theory] This ensures that words which are specific to a small group of documents and appear often in them, get a higher weight, while terms which are infrequent or too frequent in many documents, as articles for example, get a small weight.

Now that we have a vector representation of each document, we could work in the existing space and try to cluster our documents in their current form using k-Means, which will be explained later. An exemplary analysis shows that the semantic coherence of the document clusters seems to lack. [**TK:** show proof] That is due to the clustering algorithm failing to perform and facing, what is commonly known as, *the curse of dimensionality*. The curse of dimensionality states for distance based methods that "under certain reasonable assumptions on the data distribution, the ratio of the distances of the nearest and farthest neighbors to a given target in high dimensional space is almost 1 for a wide variety of data distributions and distance functions" [AHK01]. Therefore closeness between points, which is the relevance metric for the k-Means algorithm

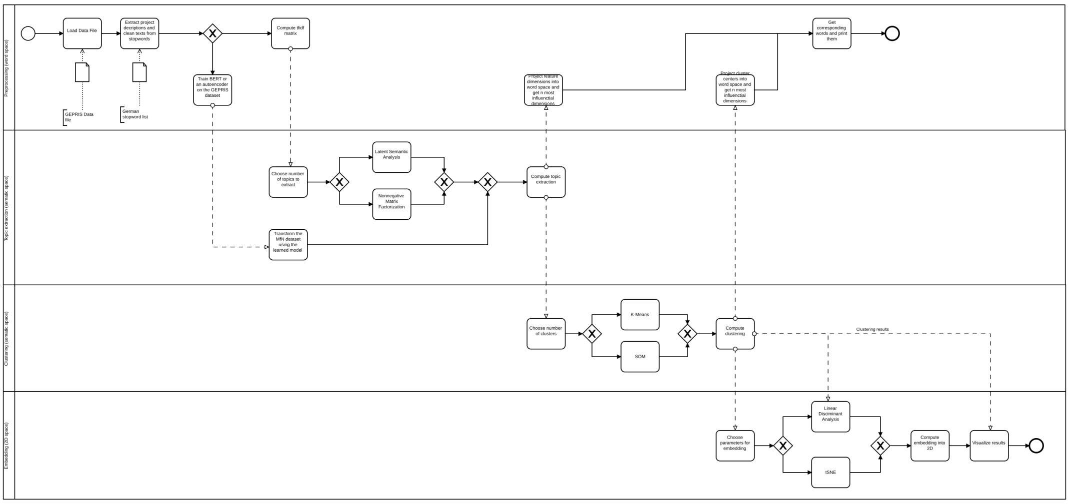

Figure 3.3: BPMN process diagram of the existing topic modeling pipeline

due to it using the Euclidian distance, becomes effectively meaningless and making it necessary to reduce the dimensionality of the vector space.

One popular method, which is often used in conjunction with Tf-Idf BOW embeddings, is the Latent Semantic Indexing (LSI), also known and henceforth referenced as Latent Semantic Analysis (LSA). A LSA operates on the premise that a vectorized corpus contains latent structures, which my correspond to topics for example. Such a topic would consist of several words which are semantically connected and therefore appear together more often than words which are not semantically similar. Adding constraints such as adjustable representational richness, which depicts sufficient parameterisation, explicit representation of both terms and documents and computational tractability for large datasets the authors decided to use a Singular Value Decomposition (SVD) [DDF+]. The SVD is closely related to Principal Component Analysis (PCA) and reduces the dimensionality of a dataset by removing the dimensions with the least variance, effectively projecting the vector space onto the subspace with the highest variance and therefore the most information contained. Applying a SVD on the corpus changes the representation of the document from being a linear combination of words into being a linear combination of latent topics. This representation is now usable for most other methods such as clustering due to its smaller dimensionality. The existing pipeline uses a k-Means algorithm to discover clusters and classify the documents as a next step. Finally, in order to visualize the high dimensional topic space in 2D a linear discriminant analysis is used using the clustering as labels. [**TK:** Connections to BPMN]

3.3 Document embedding

3.3.1 A short survey of document embedding techniques

Since 1972, the year when the Idf measure was proposed for the first time, [Rob04] a number of other techniques appeared, which are able to vectorize documents in a corpus.

Another popular technique was published by Blei et al. [BNJ03] in 2003. *Latent Dirichlet Allocation* is a hierarchical Bayesian model, which describes documents as a finite mixture of latent topics, while topics are an infinite mixture of latent topic probabilities. The LDA therefore performs the embedding and the topic extraction step at once.

Le and Mikolov [LM14] proposed *Paragraph vectors* almost a decade later using the newest advances in neural networks. This technique, also known as *Doc2Vec*, because it expands the idea of Word2Vec [MSC$^+$] to documents, utilizes a shallow neural network to run over each document with a sliding window and predict a token in this window using the other tokens and a paragraph id as a special token as context. Using a standard backpropagation algorithm to train the weights of the network the final paragraph vector consists of the weights which are used for the paragraph id. The intuition is that the paragraph vector acts as an additional storage for context information and since the connected paragraph ID is unique for each document it contains semantic information for the entire document. Choosing a low dimension as an embedding dimension also corresponds to the embedding and topic extraction step at once, but the authors recommend an embedding dimensionality of at least 100.

A rather new method was presented by Wu et al. [WYX$^+$18] using a new distance metric called *Word Mover's distance*(WMD). This metric uses pretrained word vectors and word alignment in order to compute more meaningful distances. Because the computation of this metric is quite expensive, Wu et al. develop an approximative kernel which embeds a corpus into a vector space using the WMD, which can be used instead of computing the full kernel with all the training data.

Another approach would be to not train a model on the specific dataset, but rather use a model which was pretrained on a huge and very general dataset. One of the state-of-the-art techniques for that is BERT [DCLT18]. Devlin et al. present a new model architecture based on the popular Transformer model [VSP$^+$17] and train it in the first version on a concatenated corpus of BookCorpus and the English Wikipedia ($3,3 \cdot 10^9$ words in total). Having such a huge amount of data as context knowledge one is now able to train another model for downstream tasks on top of BERT and utilize the knowledge extracted from the corpus in a transfer learning fashion. It is also possible to extract the raw document embeddings from BERT directly, but the sequence length is capped to 512 characters.

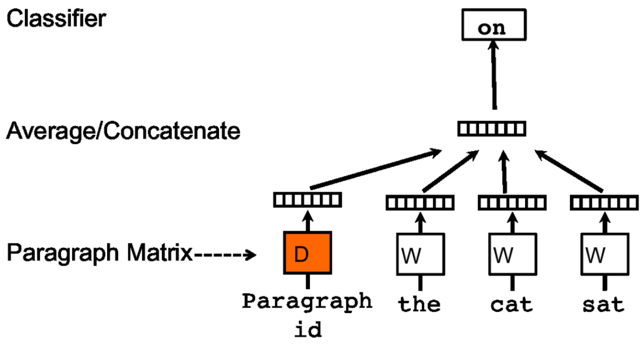

Figure 3.4: Visualization of a training step of a Doc2Vec network [WYX+18]

Technique	Parameters	Max. document length	Type
Tf-Idf BOW	0	unlimited	Probabilistic
Latent Dirichlet Allocation	1	unlimited	Probabilistic
Doc2Vec	8	unlimited	NN
Word mover's embedding	1	unlimited	Kernel method
BERT	8	512 characters	NN

Table 3.1: Table summarizing the key features of different document embedding techniques

3.3.2 Selection of a document embedding technique

Summarizing the previously discussed methods by three of their main characteristics - number of hyperparameters, maximum processable document length and type of model results in Table 3.1.

The model is now selected by exclusion. Since our database contains documents which are longer than 512 tokens and each token has a length of at least 1 character, BERT is eliminated as a potential document embedding technique. It would be possible to take word embeddings from BERT and average them in order to get a document embedding as it was proposed and further developed in [DBVCDD16] for Word2Vec embeddings, but there was no scientific or non-scientific literature that suggested that this works for the case of BERT embeddings. [**TK:** Why is BERT different?] Furthermore the previous literature analysis showed that there is not a lot of work done for explaining probabilistic models or models utilizing kernel tricks, therefore TF-Idf BOW, the LDA and the Word Mover's embedding are not of interest in this case. Only Doc2Vec remains, supporting both an unlimited document length and being of type 'NN' and therefore potentially being able to support more explainability techniques which may be developed in the future.

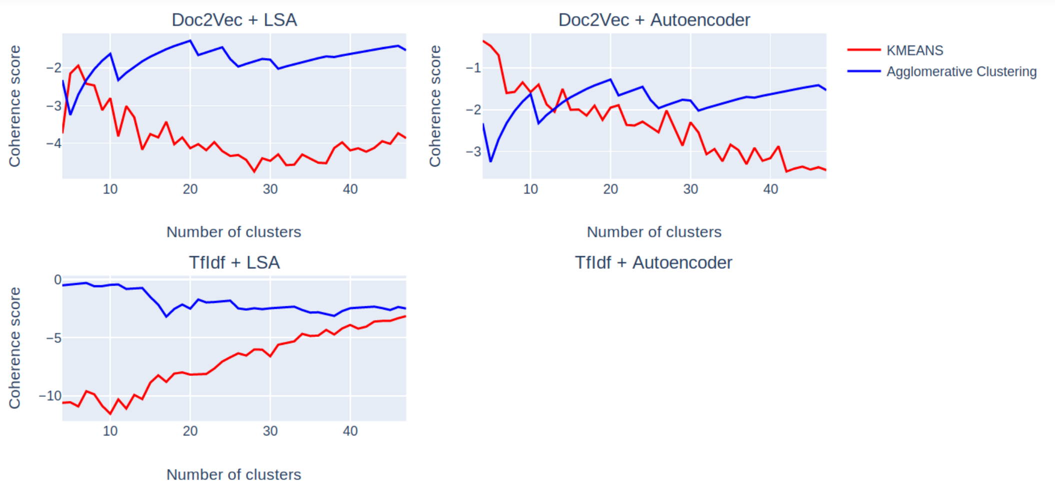

Figure 3.5: Graph showing the quality of the topic modeling while varying the embedding, topic extraction and clustering model

3.4 Topic extraction

3.4.1 Assessing the quality of the topic modeling using topic coherence

3.4.2 Explainability technique: Top words

Looking at Figure 2.6 shows that local instance explanations as a strategy to explain the output of an unsupervised embedding algorithm are most prevalent among all sourced publications. Factoring in the two questions from Table 1.1 which deal with the most prominent topics and the similarity between projects and clusters I will rank the input features to the document embedding and topic extraction step for each document and cluster.

Assume that I have a document described as a vector $c \in \mathbb{R}^k$ in the latent topic space. In order to show the most important features in the document the idea is to go backwards. Taking the vector in the latent space firstly the inverse dimensionality reduction is applied in order to transform the vector into the embedding space. Both the LSA and the autodecoder approach have this opportunity, but they differ greatly in quality of this reconstruction. Since the LSA is a linear method, the back projection yields all documents on a hyperplane, while the autodecoder is able to minimize the reconstruction loss through its inherent nonlinearity.

Now that the document vector is in its embedding space we will make use of a special ability of the Doc2Vec model. As described in the previous section the model does not only train document vectors, but it also generates word embeddings in the same space. The revolution the Word2Vec model, as a base of the Doc2Vec model, presented was that the generated embeddings and their relations to eachother encoded semantic relations. Although there is no literature on this, it is a hypothesis that this behaviour also applies

"Evolution"	"Diversität"
1. evolutionären	1. Artenzusammensetzung
2. evolutionäre	2. Biodiversität
3. evolutionärer	3. Taxa
4. phylogenetische	4. taxonomisch
5. Artbildung	5. Lebensräumen

Table 3.2: Table showing the top five similar words for two queries by word

'Ambitionierte Amateure' - Europäische Filmclubs in den langen 1960er Jahren	Netzwerke im europäischen Handel des Mittelalters
1. Kulturpraxis	1. Opportunitätskosten
2. Kulturzentren	2. Diskursteilnehmer
3. alltagsweltlich	3. evoluieren
4. pain	4. schloss
5. Ceuta	5. Staphylococcen

Table 3.3: Table showing the top five similar words for two queries by document

to the embedding of document and word vectors into one space. This leads to the possibility of describing a document by its nearest word vectors. An exemplary analysis shows that there seems to be a valid semantic structure in the relations between tokens and between tokens and documents. As a German native speaker it is easy for me to verify that the word queries in Table 3.2 are indeed semantically well connected. The results of the document queries in Table 3.3 on the other hand are hard to verify since most documents are very specific, scientific texts. I picked two projects and their top words which I was able to understand without relying on external information. The first three top words are indeed well connected to the topic of the corresponding project, but the last two ones seem to be off. This exemplary analysis speaks in favor of the hypothesis that there are indeed semantic connections document vectors and word vectors.

[**TK:** Proof that the word vectors make sense and the doc vectors also make sense]

Keeping the gist of this strategy the first explainability technique I propose id the *Top words* method.

3.5 Clustering

As described in the beginning of this chapter a K-Means clustering would now classify the documents in the latent topic space. A problem that this approach poses is that the assumption that the Euclidian distance (EuD), which is the inert similarity measure of the K-Means algorithm [**TK:** Cite!] is meaningful in our vector space may not be true. [**TK:** Cite?] Another distance measure which may may encode more semantic meaning could be the previously mentioned *Word Mover's Distance*. Since it uses the generated word embeddings and their order in the document to compute distances, it may be more suited for comparing texts and subsequently also yield better results for the clustering. This weighs even heavier if the corpus is vectorized as a sparse matrix since the Euclidian distance, as described in chapter 1, looses its meaning . In our case, having embeddings from a Doc2Vec model, we don't have to deal with this additional problem, but the question remains if the two distance measures differ on our dataset. Comparing the distances between documents generated by both the Word Mover's distance and the Euclidian distance in Figure 3.6 it is apparent that there is indeed a difference. Therefore it is worth investigating if using this information improves the clustering results.

Inspired by Liu et al. [LHLH18] I chose a hierarchical clustering approach, specifically *Agglomerative Clustering*, as a contending method to the K-Means algorithm. This method works bottom-up since in the beginning it considers every data point to be its own cluster. Now in every step two clusters are merged which minimize a given linkage metric. The distance calculations between points are performed by a lookup in a precomputed distance matrix which enables the usage of any given distance metric. Doing this until only one cluster remains, creates a binary tree, which describes the hierarchy of the data given the used metric.

3.5.1 Explainability technique: Cluster topography

3.6 Reduction into 2D

3.6.1 Explainability technique: Linearization

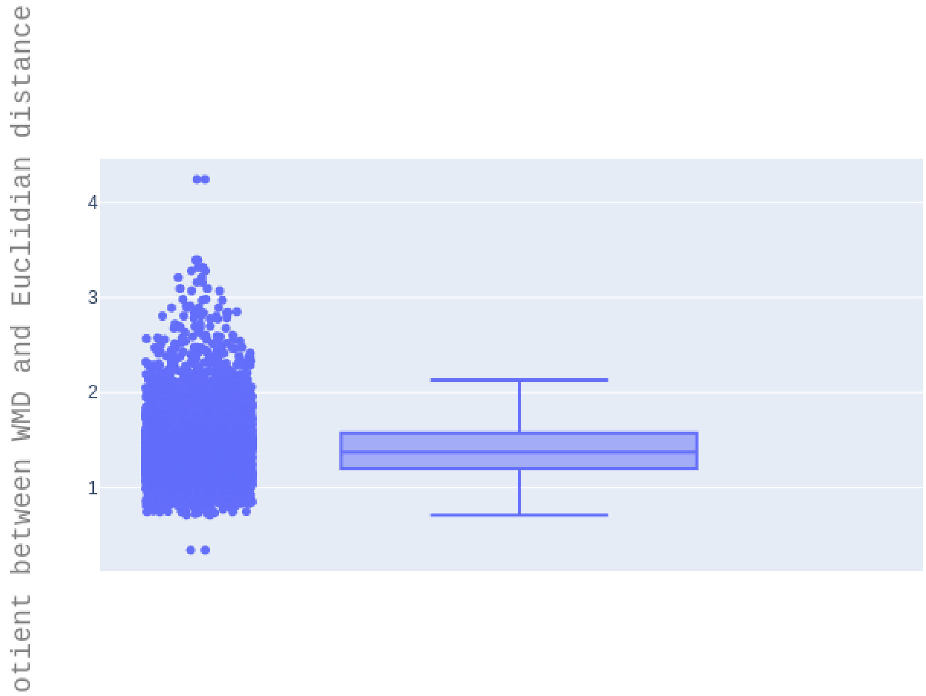

Figure 3.6: Boxplot showing the distribution of the quotients between WMD and EuD for all documents

4 Validation

4.1 Setup

4.2 Cognitive Walkthrough

5 Conclusion

5.1 Discussion

5.2 Outlook

- Die Zusammenfassung sollte das Ziel der Arbeit und die zentralen Ergebnisse beschreiben. Des Weiteren sollten auch bestehende Probleme bei der Arbeit aufgezählt werden und Vorschläge herausgearbeitet werden, die helfen, diese Probleme zukünftig zu umgehen. Mögliche Erweiterungen für die umgesetzte Anwendung sollten hier auch beschrieben werden.

Bibliography

[AHK01] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the Surprising Behavior of Distance Metrics in High Dimensional Space. In Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, Jan Van den Bussche, and Victor Vianu, editors, *Database Theory — ICDT 2001*, volume 1973, pages 420–434. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[BNJ03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[DBVCDD16] Cedric De Boom, Steven Van Canneyt, Thomas Demeester, and Bart Dhoedt. Representation learning for very short texts using weighted word embedding aggregation. *Pattern Recognition Letters*, 80:150–156, September 2016.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, October 2018.

[DDF$^+$] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, page 17.

[Den17] Arthur (New York University) Denny, Matthew (Penn State University); Spirling. Replication Data for: Text Preprocessing For Unsupervised Learning: Why It Matters, When It Misleads, And What To Do About It, 2017.

[DFG] DFG - GEPRIS. https://gepris.dfg.de/gepris/OCTOPUS?task=showAbout.

[GCJC] Yuxia Geng, Jiaoyan Chen, Ernesto Jimenez-Ruiz, and Huajun Chen. Human-centric Transfer Learning Explanation via Knowledge Graph. page 4.

[GMPB16] Yash Goyal, Akrit Mohapatra, Devi Parikh, and Dhruv Batra. Towards Transparent AI Systems: Interpreting Visual Question Answering Models. *arXiv:1608.08974 [cs]*, August 2016.

[Int] Introducing the Museum für Naturkunde in Berlin. https://pro.europeana.eu/post/introducing-the-museum-fur-naturkunde-in-berlin.

[IST+18] Tomoki Ito, Hiroki Sakaji, Kota Tsubouchi, Kiyoshi Izumi, and Tatsuo Yamashita. Text-Visualizing Neural Network Model: Understanding Online Financial Textual Data. In Dinh Phung, Vincent S. Tseng, Geoffrey I. Webb, Bao Ho, Mohadeseh Ganji, and Lida Rashidi, editors, *Advances in Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, pages 247–259. Springer International Publishing, 2018.

[KLHK19] D. Kim, W. Lim, M. Hong, and H. Kim. The Structure of Deep Neural Network for Interpretable Transfer Learning. In *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 1–4, February 2019.

[LHLH18] Ninghao Liu, Xiao Huang, Jundong Li, and Xia Hu. On Interpretation of Network Embedding via Taxonomy Induction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pages 1812–1820. ACM, 2018.

[Lip16] Zachary C. Lipton. The Mythos of Model Interpretability. *arXiv:1606.03490 [cs, stat]*, June 2016.

[LM14] Quoc V. Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. *arXiv:1405.4053 [cs]*, May 2014.

[LTD+16] Lin Liu, Lin Tang, Wen Dong, Shaowen Yao, and Wei Zhou. An overview of topic modeling and its current applications in bioinformatics. *SpringerPlus*, 5(1):1608, September 2016.

[MHS17] Tim Miller, Piers Howe, and Liz Sonenberg. Explainable AI: Beware of Inmates Running the Asylum Or: How I Learnt to Stop Worrying and Love the Social and Behavioural Sciences. *arXiv:1712.00547 [cs]*, December 2017.

[MSC+] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. page 9.

[MWM18] D. L. Marino, C. S. Wickramasinghe, and M. Manic. An Adversarial Approach for Explainable AI in Intrusion Detection Systems. In *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pages 3237–3243, October 2018.

[PFMM] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. Systematic Mapping Studies in Software Engineering. page 10.

[Piv] Pivoted document length normalisation | RARE Technologies. https://rare-technologies.com/pivoted-document-length-normalisation/.

[Rob04] Stephen Robertson. Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation*, 60(5):503–520, October 2004.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[WYX+18] Lingfei Wu, Ian E. H. Yen, Kun Xu, Fangli Xu, Avinash Balakrishnan, Pin-Yu Chen, Pradeep Ravikumar, and Michael J. Witbrock. Word Mover's Embedding: From Word2Vec to Document Embedding. *arXiv:1811.01713 [cs, stat]*, October 2018.

[ZYL+18] Quanshi Zhang, Yu Yang, Yuchen Liu, Ying Nian Wu, and Song-Chun Zhu. Unsupervised Learning of Neural Networks to Explain Neural Networks. *arXiv:1805.07468 [cs]*, May 2018.