The Design of an Interactive Topic Modeling Application for Media Content

Laura Ham

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.
Espoo 10.7.2020

Supervisor

Prof. Dr. Antti Oulasvirta

Advisor

Dr. Luis A. Leiva

Aalto University
School of Science

Aalto University
School of Science

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Laura Ham

Title The Design of an Interactive Topic Modeling Application for Media Content

Degree programme ICT Innovation

Major Human-Computer Interaction and Design **Code of major** SCI3020

Supervisor Prof. Dr. Antti Oulasvirta

Advisor Dr. Luis A. Leiva

Date 10.7.2020 **Number of pages** 73+28 **Language** English

Abstract

Topic Modeling has been widely used by data scientists to analyze the increasing amount of text documents. Documents can be assigned to a distribution of topics with techniques like LDA or NMF, that are related to unsupervised soft clustering but consider text semantics. More recently, Interactive Topic Modeling (ITM) has been introduced to incorporate human expertise in the modeling process. This enables real-time hyperparameter optimization and topic manipulation on document and keyword level. However, current ITM applications are mostly accessible to experienced data scientists, who lack domain knowledge. Domain experts, on the other hand, usually lack the data science expertise to build and use ITM applications.

This thesis presents an Interactive Topic Modeling application accessible to non-technical data analysts in the broadcasting domain. The application allows domain experts, like journalists, to explore themes in various produced media content in a dynamic, intuitive and efficient manner. An interactive interface, with an embedded NMF topic model, enables users to filter on various data sources, configure and refine the topic model, interpret and evaluate the output by visualizations, and analyze the data in wider context. This application was designed in collaboration with domain experts in focus group sessions, according to human-centered design principles.

An evaluation study with ten participants shows that journalists and data analysts without any natural language processing knowledge agree that the application is not only usable, but also very user-friendly, effective and efficient. A SUS score of 81 was received, and user experience and user perceptions of control questionnaires both received an average of 4.1 on a five-point Likert scale. The ITM application thus enables this specific user group to extract meaningful topics from their produced media content, and use these results in broader perspective to perform exploratory data analysis.

The success of the final application design presented in this thesis shows that the knowledge gap between data scientists and domain experts in the broadcasting field has been filled. In bigger perspective; machine learning applications can be made more accessible by translating hidden low-level details of complex models into high-level model interactions, presented in a user interface.

Keywords Interactive Machine Learning, Topic Modeling, Human in the loop, Data visualization, User interface design

Preface

This thesis marks the last chapter of my two years Master's Program in Human Computer Interaction and Design at EIT Digital. I had the pleasure of studying at Twente University in the Netherlands, and at Aalto University in Finland. In this two-year journey I have not only acquired academic and professional experience, but I also expanded my network with fellow motivated innovators. I want to thank everyone from EIT Digital, Twente University and Aalto University who have supported me in these two years.

There are several people I would like to thank for their support to this thesis project. First of all, I would like to thank my supervisor Prof. Antti Oulasvirta and my advisor Dr. Luis Leiva from Aalto University for providing me guidance throughout the project. Your prompt and elaborate feedback helped me shape this thesis.

Secondly, I would like to thank Lauri Mikola and Eija Moisala for giving me the opportunity to perform this research at Yle. I want to thank everyone from the Smart Data and Audience Insights team at Yle, who welcomed me warmly into the team. It was a pleasure to work with all of you. Special thanks also to Elina Kuuluvainen, Terhi Upola and Jarno Kartela who have been of great inspiration to the research, design and development of this project. Additionally, I would like to thank all the participants that joined the user study for this thesis.

Last but not least, I would like to thank my friends and family who supported and encouraged me throughout this process.

Otaniemi, 31.8.2020

Laura Ham

Contents

List of Figures

List of Tables

Symbols and abbreviations

Symbols

w	A single word or term
d	A single document
z	A single latent topic
m	The number of documents
n	The number of words or terms
k	The number of topics
$\mathcal{N}$	The set of words or terms in a document
$\mathcal{M}$	The set of documents
$\mathbf{A} \in \mathbb{R}^{m \cdot n}$	Document-term matrix
$\mathbf{U} \in \mathbb{R}^{m \cdot k}$	Document-topic matrix (LSA)
$\mathbf{V} \in \mathbb{R}^{k \cdot n}$	Topic-term matrix (LSA)
$\mathbf{W} \in \mathbb{R}^{m \cdot k}$	Document-topic matrix (NMF)
$\mathbf{H} \in \mathbb{R}^{k \cdot n}$	Topic-term matrix (NMF)

Abbreviations

BOW	Bag-of-words
EM	Expectation Maximization
HCD	Human-Centered Design
ITM	Interactive Topic Modeling
LDA	Latent Dirichlet Allocation
LSA	Latent Semantic Analysis
NMF	Non-negative Matrix Factorization
NPMI	Normalized Pointwise Mutual Information
PCA	Principle Component Analysis
pLSA	Probabilistic Latent Semantic Analysis
SVD	Singular Value Decomposition
tf-idf	Term Frequency-inverse Document Frequency
TM	Topic Modeling
t-SNE	T-distributed Stochastic Neighbor Embedding

1 Introduction

Broadcasting companies have large amounts of media content, and the production keeps increasing. Extracting meaningful insights from these large text corpora requires efficient analysis methods. Data-driven modeling techniques like clustering or topic modeling for automatic theme discovery are often used. Topic modeling (TM) is a growing research field that advances from data mining, machine learning, data analytics, data visualization and user-machine interaction.

Topic modeling infers latent structures of large document collections by automatically coding them into a smaller number of semantically meaningful categories. Topic model algorithms are built upon the presumption that semantics are relational, and thus assume that documents contain similar words if they share a latent topic. Co-occurrence patterns of word bags are extracted by these algorithms, ignoring regular natural language complexities such as syntax and location. The procedure requires, in contrast to traditional approaches of text analysis, only minimal human intervention, and is thus scalable and efficient in use.

Common TM approaches take an input in the form of a term-document matrix representation of documents via a bag-of-words model. Probabilistic or matrix factorization methods typically represent topics by a weighted combination of keywords and individual documents by a weighted combination of topics. A key characteristic is that these extracted topics are latent, and thus can be best interpreted by humans.

1.1 Problem Definition

Previous research shows that existing topic models have several shortcomings. For example, the automatically discovered topics can be hard to interpret and do not always make sense. Extracting too many or too few topics leads to too general or too specific results [19]. Interactive Topic Modeling (ITM) has been introduced recently, which incorporates human expertise in the modeling process [23]. ITM applications allow users to refine extracted topics on topic, keyword and document level. These applications are typically used by data scientists, who are experienced in natural language processing and topic modeling on a technical level. These experts often lack domain knowledge about the data and its representation in bigger context. Domain experts in the context of broadcasting, like journalists and media data analysts, have this knowledge about the produced and consumed media content, but usually lack the technical data science skills to develop and use complex models for topic extraction. Topic models and other machine learning techniques are currently not optimally used because of the combination of gaps in each other's knowledge and skills. The knowledge gap is present between data scientists, visualization researchers and domain experts, mainly because patterns of thinking and strategies for solving problems differ significantly [49]. Moreover, there is information loss in the interdisciplinary communication because domain experts find it hard to articulate their problems and tasks [46], [57]. Regarding media data analysis, broadcasting agencies are missing out on opportunities, such as trend discovery based on how themes in produced content are consumed in different audience groups.

1.2 Research Goal

The goal of this thesis project is to fill this gap of lacking domain knowledge of data scientists and data science skills of domain experts by combining the expertise of both groups into one application which is accessible to non-technical domain experts. The application should enable domain experts to find and analyze latent topics in media content, using interactive topic modeling. Hiding complex model computations in the background and showing the model's data input, output and refinements in a simple interactive interface should make ITM accessible for non-technical users. This thesis researches the hypothesis stating that *'An Interactive Topic Modeling Application accessible to non-technical users bridges the knowledge gap between data scientists and domain experts'*. This research is applied in the domain of multimedia content production, but is aimed to discover design and development methods which can be applied in the bigger perspective of interactive topic modeling and machine learning accessibility.

1.3 Research Methodology

Developing an ITM application requires expertise from data scientists as well as domain experts. Domain experts, the end-users of the application, are for example journalists or media data analysts. The research, design and development methodology of this project is inspired by principles of the human-centered design (HCD) process [24], which considers expertise and requirements from multiple stakeholders. In addition, this approach aims to make interactive systems more usable by focusing on system usage and human ergonomics. The widely-adopted framework provides requirements and recommendations for designing interactive systems. It places the needs of end-users at the heart of the design and development process, which consists of four phases (Figure 1): (1) *Identifying the use context*; (2) *Identifying the user requirements* (through desk research and focus group sessions); (3) *Generate and prototype solutions* (through participatory design, concept testing and desirability studies) and (4) *Evaluate solutions through testing and measuring*. Functional testing, including User Acceptance Testing, of the application is done iteratively in the design cycles. After the last cycle, an in-depth user evaluation study is conducted, assessing the usability and user experience. Participants are given a modeling task, and quantitative and qualitative data is gathered using a thing-aloud protocol, post-task semi-structured interviews and questionnaires.

The design and development process has an iterative nature, so that the interactive software product will be developed incrementally. According to the ISO 9241-210:2019 guidelines, the human-centered approach follows at least the following principles:

1. The design is based upon an explicit understanding of users, tasks and environments;

2. Users are involved throughout design and development;

3. The design is driven and refined by user-centered evaluation;

4. The process is iterative;
5. The design addresses the whole user experience;
6. The design team includes multidisciplinary skills and perspectives.

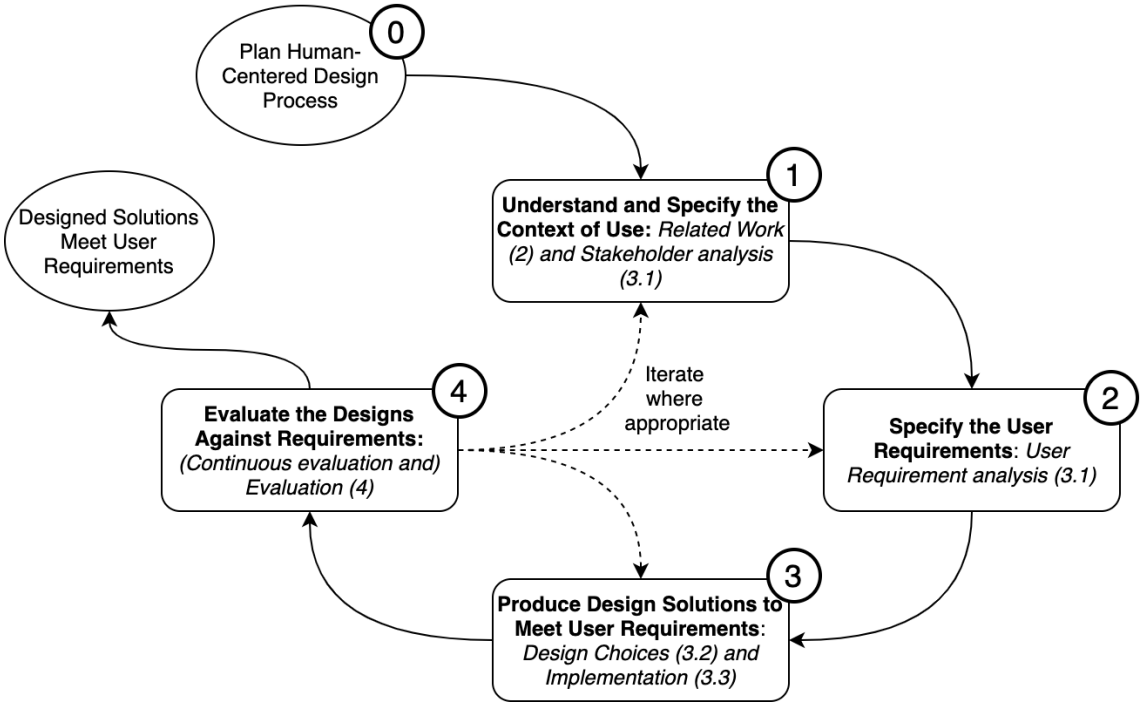

Figure 1: The iterative human-centered design (HCD) process (adopted from [24]) that is used as research, design and development methodology in the thesis. The italic numbers in brackets indicate the section which covers the content of the steps.

1.4 Context: Finland's national public broadcasting company Yle

This research is done in the context of Finland's national broadcasting company Yle[1]. In 2019 alone, Yle had 16,400 hours of recorded media on their online streaming platform Yle Areena[2], 18,900 hours of programs on their four television channels and 47,500 hours of radio programs. 96% of the Fins are reached weekly.

Broadcasting companies like Yle have various use cases with these large amounts of produced and consumed media content. Use cases range from simple theme discovery to doing comparative trend analysis on produced and consumed article content clustered by the underlying topics. Domain experts from Yle considered in this thesis typically lack data science knowledge. Close collaboration was established during the iterative, human-centered design approach which was adopted in the

[1] https://yle.fi/
[2] https://areena.yle.fi/tv

research and development of the application. Media planners from various multimedia departments of Yle are still using the interactive application presented in this thesis on their own media datasets.

1.5 Structure of Thesis

The main part of this thesis is structured according to the HCD framework. It starts with specifying the context of use in Section 2. A literature study is conducted on relevant topics to determine the landscape of Interactive Topic Modeling methods and application design. First, Topic Modeling and interpretation techniques are discussed, followed by how existing Interactive Topic Modeling techniques take the human in the modeling loop and what the capabilities are of those current applications.

This forms the basis for the next step in the HCD cycle: specifying the user requirements and making design choices. The former step consists of a Requirements Analysis, presented in Section 3.1, to identify different user groups, as well as their motivations and challenges. These user requirements are essentially features and attributes the product should have and tells how it should perform. Section 3.2 covers the design choices that are made by combining these requirements with findings from related work. The final implementation presented in Section 3.3 is built upon these design choices.

This is followed by an evaluative user study of the application, described in Section 4. Section 5 discusses the results, limitations of the application and research approach, and suggests future work. Finally, Section 6 summarizes and concludes this project.

2 Related Work

2.1 Topic Modeling

Topic modeling is a machine learning method for grouping a set of documents according to their semantic themes. This text mining technique identifies co-occurring keywords to summarize large collections of textual data. Topic modeling is used to discover hidden themes in documents, annotate documents with topics and organize large amounts of unstructured text data. All topic models have two basic assumptions:

1. Each document consists of a mixture of topics, and
2. Each topic consists of a collection of words.

There is no prior knowledge required about what the documents contain. Topic models typically reduce the dimensionality of a set of words in documents into a smaller set of interpretable and meaningful topics. Topic modeling methods are according to the first assumption mixed-membership models, unlike other unsupervised methods like K-means clustering or Naive Bayes. Documents can thus have a distribution under several identified topics. Additionally, words can be associated with multiple topics. Figure 2 visualizes this concept.

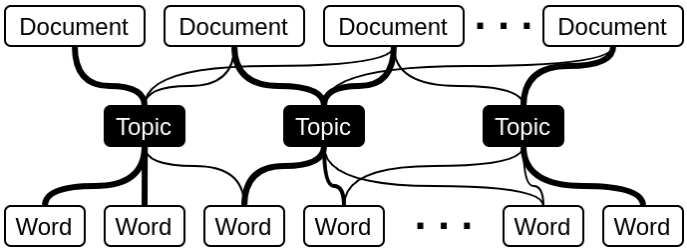

Figure 2: The concept of topic modeling. Documents are associated with one or more topics, which are represented by multiple words. Line thickness indicates the degree of a topic being represented in the document, and words in topics.

Another characteristic of topic modeling is that derived topics are latent. This means that the results are hard to be interpreted by a machine, since they describe the semantics of groups of documents. However, the high interpretability of modeling results means that these results are useful for exploring large datasets by humans.

This subsection presents different modeling methods to extract topics from textual data, but first the difference between soft en hard clustering and topic modeling is explained.

2.1.1 Clustering

Unsupervised clustering is typically a technique to discover groups of similar samples in a collection of unlabeled data. It tries to find a structure within a dataset. In terms of unsupervised document clustering, hard clustering results in a set of clusters each containing a set of documents, where documents can belong to a single cluster

only. These unsupervised clustering techniques use *unigram models*. Here, each word is assumed to be drawn from the same distribution, and does not model documents dealing with a mixture of topics.

However, in real-life we often do not want to assign a document to a single cluster, since multiple topics can be discussed in one document. This asks for *mixture of unigram models* or *mixed-membership models*; soft clustering techniques where documents are simultaneously assigned to belong to several topics, and where topic distributions vary over the documents [39].

In contrast to hard clustering, soft clustering allows data to be represented as weighted combinations of clusters in terms of their proximity to each cluster. Thus, soft clustering is related to topic modeling; documents can belong to multiple clusters, as a distribution by weights. There is however a difference between soft clustering and topic modeling: soft clustering does not consider the semantics of words, documents and clusters. It only takes that relatedness of documents to each cluster into account, while topic modeling considers both.

Topic modeling considers semantics, because in textual data there is often a difference between the actual text (lexical level) and the intention or meaning (semantic level) of it. In addition, natural language data may contain *polysems* (i.e. a word that has multiple senses and multiple types of usage in different context) and *synonyms* (i.e. different words with the same meaning or referring to the same topic), which forms a problem for machine learning methods. Hard and soft clustering methods cannot solve this challenge without considering semantics. Topic modeling is thus usually preferred over these traditional clustering methods for the discovering latent structures in document sets, because it considers semantics and has the mixed-membership property. Various topic modeling methods have been introduced, each one having its own advantages and disadvantages. The next four subsections introduce the most common techniques.

2.1.2 Latent Semantic Analysis (LSA)

Latent semantic analysis (LSA, also known as latent semantic indexing, LSI), is one of the first known traditional methods for topic modeling. It was developed in the late 1980s by Deerwester et al. [14] as a technique to improve information retrieval [16]. This method is based on finding latent document structures by linear algebra instead of using straightforward document term structures and tf-idf. Documents are represented by 'hidden' semantic concepts, not merely by the terms occurring in them. LSA is a dimensionality reduction technique, which maps documents to a reduced dimensionality. This dimensionality reduction is performed by singular value decomposition (SVD) of the document-term matrix, retaining the components with largest variance. The decomposition of the document-term matrix results in two singular and one diagonal matrix:

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}} \tag{1}$$

in which $\mathbf{A} \in \mathbb{R}^{m \cdot n}$ is the document-term matrix, $\mathbf{U} \in \mathbb{R}^{m \cdot k}$ the document-topic matrix, $\Sigma \in \mathbb{R}^{k \cdot k}$ a diagonal topic importance matrix and $\mathbf{V}^{\mathrm{T}} \in \mathbb{R}^{k \cdot n}$ the topic-term

matrix. m, n and k denote the number of documents, terms and topics respectively. This is also visualized in Figure 3.

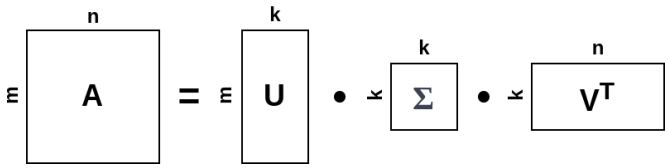

Figure 3: Singular Value Decomposition of the document-term matrix ($\mathbf{A}$) into the document-topic matrix ($\mathbf{U}$), a diagonal topic importance matrix (Σ) and the topic-term matrix ($\mathbf{V}^{\mathrm{T}}$). The number of documents, terms and topics are denoted by m, n and k respectively.

The original vectors in the high-dimensional document-term matrix are sparse, but the resulting corresponding low-dimensional latent vectors are typically not sparse. This makes it possible to compute association values between document pairs, even if there are no common words. The idea of LSA is that semantically similar words are mapped in the same direction in the latent space.

Although LSA finds latent semantic document structures and overcomes the problem of having polysems and synonyms across documents, this method is not often used in real-world topic modeling applications. The main reason is that the model has the possibility to assign negative weights, which makes the resulting keywords and topics hard to interpret. In addition, as explained, SVD only learns the span of the topics, not the actual topics themselves. For real-life applications, recovering the documents' distribution over topics is desirable over just having the span, so that users can explore the document set by the actual topics, and use the topics on documents outside the training dataset. For these reasons, in the domain of topic modeling, more recent methods focus on probabilistic modeling such as probabilistic LSA (pLSA) and Latent Dirichlet Allocation (LDA).

2.1.3 Probabilistic Latent Semantic Analysis (pLSA)

pLSA approaches the same problem as LSA, but fits an underlying generative probabilistic model to the observed data using expectation maximization (EM) [21]. A mixture decomposition is derived from a latent class model, where overfitting is prevented by maximum likelihood model fitting. A corpus is modeled as the mixture model, where each word in a document is a sample from the model. The latent topics are represented by multinomial random variables, which are the mixture components of the model.

pLSA requires only one hyperparameter to be set before the start of the modeling process: the number of topics k to extract. The modeling procedure is then as follows. A document d is selected with probability $P(d)$. A latent class z is picked with probability $P(z|d)$ from a multinomial distribution. Then, a word w is generated with probability $P(w|z)$, also from a multinomial distribution. The only observed result is the pair (d, w); the latent class z is not observed. It is assumed that the

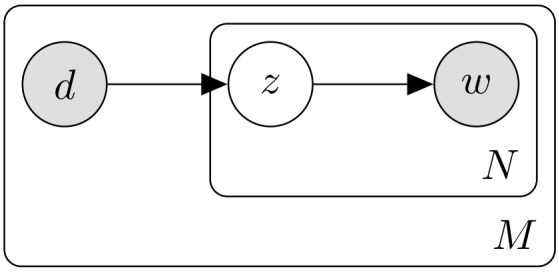

Figure 4: Graphical model representation of pLSA [25]. Document d is drawn from the document set $\mathcal{M}$ with probability $P(d)$. A latent class z is drawn from a multinomial distribution with probability $P(z|d)$. A word w from the word set $\mathcal{N}$ is generated from a multinomial distribution with probability $P(w|z)$.

generated pairs (d, w) are independent, and that the words w are independent from the documents d.

The generative process is visualized in the graphical plate diagram in Figure 4. The outer plate represents a set of $\mathcal{M}$ observed documents d. This is represented by a mixture of latent topics z, from which terms w are drawn for each document in the word set $\mathcal{N}$ (represented by the inner plate).

Using the EM algorithm, the likelihood of the data is maximized given the model. In the expectation step, the posterior probabilities of the latent classes z are estimated, while the parameters are updated to maximize posterior probability in the maximization step. The resulting observed document-term pairs may be generated by multiple topics, and thus each document is represented by a topic mixture.

Although pLSA tackles some shortcomings of LSA, this method could still be improved. For example, the model is likely to overfit since the number of parameters grows linearly with the number of documents. In addition, there is no possibility to assign topic probabilities to documents outside the training corpus, because there are no parameters to model the probability of a document (pLSA is a generative model of the document it is modeled on, but is not a generative model of unseen documents). This makes the model less attractive for real-world use cases. Both drawbacks can be overcome by applying Latent Dirichlet Allocation (LDA), which extends pLSA with a generative model by using priors for document-topic and topic-term distributions, lending itself to better generalization and possibility to model documents outside the training corpus.

2.1.4 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is a widely used topic modeling method. LDA was introduced by Blei, Ng and Jordan in 2003 [7] as a generative, probabilistic model. Each topic is modeled as a probability distribution over words, given a predefined number of topics. Likewise, LDA models each document as a probability distribution over the topics.

Suppose there are $\mathcal{M}$ documents with n words per document. A graphical representation of the LDA model is shown in Figure 5. Here, the set of documents

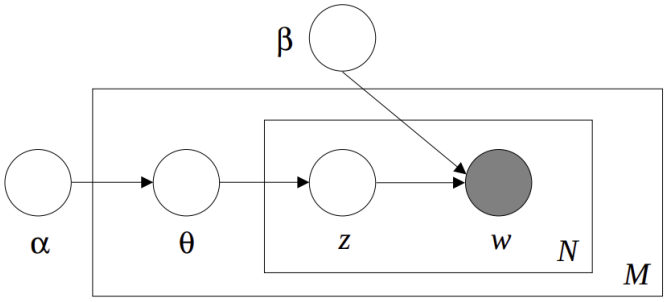

Figure 5: Graphical model representation of LDA [7]. Similar notation as in Figure 4 is used. The parameters α and β are sampled before the modeling process. The document-level variable θ_d denotes the topic distribution per document d. The word-level variables z_{d_n} (topic associated with the n-th word in document d) and w_{d_n} (the specific word) are sampled once per word per document.

$\mathcal{M}$ and words $\mathcal{N}$ are represented as outer and inner plates respectively. The nodes represent the model parameters. White nodes are latent variables, meaning that they are not directly observable, but inferred from other variables that are observed in the model. The gray node, representing the variable w, is the only observed variable, denoting the words shown in documents. The parameters α and β are sampled (tuned by users) once in the modeling process when creating a corpus, and are thus drawn outside the outer plate. α is a fixed uniform Dirichlet prior on the *per-document-topic* distributions. β is a deterministic parameter of the Dirichlet prior distribution on the *per-topic-term* distribution. The document-level variables θ_d are sampled once per document. θ_d is the topic distribution per document d. Finally, the word-level variables z_{d_n} and w_{d_n} are sampled once per word per document. z_{d_n} is a single topic associated with the n-th word in document d. w_{d_n} is the specific word (i.e. term). All in all, Figure 5 visualizes the three levels of the LDA representation, where the inner plate N represents the repeated choice of topics and words within a document, and the nodes around the plate define the sampling distribution parameters.

Three hyperparameters should be set to start the generative process: k (the number of topics), α (which controls the topic mixtures of documents) and β (which controls the distribution of words per topic). A modeling step is performed by LDA as follows. First, the number of terms n in a document is sampled from a Poisson distribution. A multinomial distribution θ over k topics for each document d in $\mathcal{M}$ is then sampled from the prior Dirichlet distribution parameterized by α. Next, for each word in all words $\mathcal{N}$ in a document d, a topic z_n is sampled from the document specific topic distribution θ_d. A word is sampled from the probability distribution over the words for this sampled topic. This term w_n is sampled with probability $p(w_n|z_n, \beta)$ from the probability distribution which is multinomial, conditioned on the sampled topic z_n.

The goal of the modeling process is to find a set of topics that best describe the given set of documents. In essence, LDA approaches learning the various distributions as a statistical inference problem, where the the joint probability over the documents, terms and topics is the posterior probability that needs to be inferred. Typically,

the posterior distribution is intractable for exact inference, but multiple algorithms exist for approximating this inference. Methods include Markov Chain Monte Carlo simulation [43], Gibbs sampling [20], variational Bayes approximation [25] and likelihood maximization. A convexity-based variational algorithm was presented by Blei, Ng and Jordan [7], but has high computational complexity. Regardless of the algorithm, typical for LDA is that the topic node is sampled repeatedly within the document. Alternating the inference and parameter estimation steps maximizes the overall likelihood.

LDA has improved performance over the previously introduced models LSA and pLSA, and is used in various applications (see Section 2.3.2). Compared to LSA by topic coherence experiments (a measure indicating how semantically close words describing this topic are, i.e. topics described by words that are semantically related have a high topic coherence), LDA is better at discovering descriptive topics, while LSA performs better at creating compact semantic representations of documents and terms in corpus [55].

Nevertheless, LDA has shortcomings, mainly in terms of consistency and convergence [11]. Consistency from multiple runs indicates how stable the model output is from multiple runs in the same setting. Empirical convergence means how early the model converges from a user's perspective, contrasted to algorithmic convergence. Both model consistency and convergence are important from the user's point of view; low consistency and slow model convergence lead to low user experience. In addition, determining the optimal value of hyperparameters is hard and may lead to confusion and eventually misconception of terms, especially for non-experts. It is desirable to have model consistency and fast convergence, and avoid complicated model tuning, to ensure high user experience. An alternative topic modeling approach, Non-negative Matrix Factorization (NMF), overcomes these aforementioned problems [11].

2.1.5 Non-negative Matrix Factorization (NMF)

Like LSA, pLDA and LDA, non-negative matrix factorization is a dimensionality reduction method. But where LDA and pLSA take a probabilistic approach, NMF, like LSA, uses linear algebra principles for identifying latent structures in data. NMF is very similar to LSA, but adds a non-negativity constraint, leading to outcomes that are naturally interpretable. Paatero and Tapper [42] introduced NMF in 1994, and was first applied to environmental applications. Nowadays, NMF is applied to problems in a broad range of areas like computer vision, bioinformatics, text mining, and many more.

The core idea of NMF is as follows. Suppose a non-negative matrix $\mathbf{A} \in \mathbb{R}^{m \cdot n}$ is given. The goal of NMF is to find two matrices, $\mathbf{W} \in \mathbb{R}^{m \cdot k}$ and $\mathbf{H} \in \mathbb{R}^{k \cdot n}$, containing only non-negative values, such that

$$\mathbf{A} \approx \mathbf{WH} \tag{2}$$

Since dimensionality reduction is applied when solving (2), it is assumed that k satisfies $k < \min(m, n)$. An optimization problem can then be defined by a specific divergence or distance measure, to find the matrices $\mathbf{W}$ and $\mathbf{H}$. Various

beta divergences can be used to solve this optimization problem, for example the Frobenius norm, Kullback-Leibler (KL) divergence or Itakura-Saito (IS) distance. The Frobenius norm is a distance measure between two matrices, while KL is a distance measure between two probability functions. The Itakura-Saito distance measure reflects the perceptual similarity between the original and approximated spectrum. Algorithms to solve the optimization problem with KL are, as [60] showed, typically much slower than those using the Frobenius norm. Therefore, the most commonly used cost function is the Frobenius norm, also used in K-means clustering. The mathematical formulation of minimizing this distance measure is:

$$\min_{\mathbf{W}\geq 0,\mathbf{H}\geq 0} f(\mathbf{W},\mathbf{H}) = ||\mathbf{A} - \mathbf{W}\ \mathbf{H}||_F^2 \tag{3}$$

with the constraint all values of $\mathbf{W}$ and $\mathbf{H}$ being non-negative.

Solving the optimization problem in NMF with this Frobenius norm as divergence method is successfully used in many partitional clustering applications. It has also been shown that this method works well in topic modeling [27]. One reason why NMF performs especially well in topic modeling is that the results (matrices $\mathbf{W}$ and $\mathbf{H}$) can be seen as document-topic and term-topic results directly.

The parameters and matrices used in NMF for topic modeling can be explained as follows. The original matrix $\mathbf{A}$ represents the document-term matrix. This matrix has dimensions $m \cdot n$, where m is the number of documents and n the number of terms in the corpus. A simplification of matrix $\mathbf{A}$ is visualized in Table 1. The numbers in the matrix are the counts of the words per document. NMF decomposes this table into two smaller matrices, $\mathbf{W}$ and $\mathbf{H}$, which contain the weights and features respectively. $\mathbf{W}$ and $\mathbf{H}$ are constructed by the algorithm which takes, besides matrix $\mathbf{A}$, only one other input, k. k is the number of topics top be extracted, which will be the dimensionality of the factors $\mathbf{W}$ and $\mathbf{H}$. The core idea of NMF is to find these k vectors that are linear independent in the vector space spanned by the documents in the rows of $\mathbf{A}$, which will reveal the latent structure of the data. k will be the amount of columns in the document-topic matrix $\mathbf{W}$ and the number of rows in the topic-term matrix $\mathbf{H}$.

Matrix $\mathbf{W}$ is visualized in Table 2, this matrix consists of the m documents (rows) and k topics (columns). Each value in a row represents how much this document is related to each topic. A large value indicates a strong relation between a document and a topic. Matrix $\mathbf{H}$ (Table 3) consists of the topics and terms, thus has dimension $k \cdot n$. The values in the topic rows describe by which terms a topic can be described. In other words, $\mathbf{W}$ gives document-wise representation, while $\mathbf{H}$ gives a topic-wise representation. Large values mean that there is a strong relationship between the word and the topic. The product of $\mathbf{W}$ and $\mathbf{H}$ is then a matrix with the same shape as document-term matrix $\mathbf{A}$. Thus, each column in the product matrix is a linear combination of all column vectors in $\mathbf{W}$ with the corresponding coefficients in matrix $\mathbf{H}$. Assuming the factorization worked, the product of $\mathbf{W}$ and $\mathbf{H}$ is a reasonable approximation to the input matrix $\mathbf{A}$. A characteristic is that column vectors in $\mathbf{W}$ and $\mathbf{H}$ do not necessarily sum up to one (the columns do not have a unit L_1-norm), unlike LSA, pLSA and LDA outputs. This difference is however negligible since

diagonal scaling could be applied to manipulate the column values.

Table 1: Example document-term matrix **A**

	Term 1	**Term 2**	**Term 3**	**Term ...**	**Term n**
Document 1	1	0	0	...	2
Document 2	0	1	0	...	0
Document 3	0	1	0	...	1
Document ...	...	...	...	...	...
Document m	0	0	0	...	0

Table 2: Example coefficient (document-topic) matrix **W**

	Topic 1	**Topic 2**
Document 1	1	0
Document 2	0	1
Document 3	0	1
Document ...	...	...
Document m	0	0

Table 3: Example feature (topic-term) matrix **H**

	Term 1	**Term 2**	**Term 3**	**Term ...**	**Term n**
Topic 1	0.5	0	0	...	1
Topic 2	0	0.5	0	...	0

NMF uses the same basic principle as LSA, where topics are learned by smoothing counts to enhance weights of the most informative words in document-term matrices, which discovers relations between words and documents. Traditionally, LSA uses SVD to solve this matrix factorization problem. SVD decomposes the document-term matrix **A** into three smaller matrices, of which one is a diagonal singular value matrix (see Figure 3). The rows of two other matrices are constraint to be orthonormal eigenvectors. The algorithm tries to minimize the reconstruction error (minimizing its Frobenius norm). NMF decomposes the original matrix **A** into two matrices instead of three. Similar to SVD, NMF factorizes the document-term matrix by minimizing reconstruction error, but with the only constraint that all values in the two decomposed matrices are non-negative.

Although the objective function is the same for both NMF and LSA with SVD, the outcome of NMF is often preferred in real-life topic modeling settings because of its non-negativity constraint. The resulting matrices **W** and **H** are more interpretable, which leads to better topic understanding by users.

According to a comparative research of [55] regarding topic coherence of different methods, LDA, NMF and LSA achieve similar topic coherence. From a computational perspective, NMF is often applied in topic modeling. The underlying modeling

calculations are relatively easy and cheap compared to other approaches like LDA. In addition, NMF overcomes the limitations of hard clustering like restricting to one topic per document and learns, like SVD, the span of topics instead of discovering the latent topics [4]. The major drawback of LSA with SVD, the unintelligible resulting latent space, is also overcome by the introduction of the non-negativity constraint of NMF.

The difference between LDA and NMF is thus mainly that the former approach uses a Dirichlet prior in the generative process, which means that topics and terms are allowed to vary per document. Probability vectors of the multinomials are fixed in NMF, which means that, from a quality perspective, NMF may lead to worse mixtures and thus topic distributions. Although the slightly poorer results of NMF may be a disadvantage in some settings, for topic modeling in an intuitive user application, model consistency is at least as important. In addition, NMF is deterministic and user interactions beyond changing static things like parameters and initial values can be easily incorporated via forms of semi-supervisions.

2.2 Topic Modeling Visualization

Once latent topics are extracted from a corpus using one of the techniques explained above, the next challenge is presenting this to human users. An intuitive representation of the extracted topics, as well as the underlying model, is desired to promote understanding of this high-level statistical tool and its results. Especially in Interactive Topic Modeling settings, where the user is asked to improve modeling results by manipulation, good understanding is essential. This subsection covers how topic model results are presented in related work. In addition to different types of (graphical) visualizations, methods of topic labeling and word ranking are discussed. How user interactions can be incorporated in topic modeling is discussed in subsection 2.3.

2.2.1 Visualizations in existing applications

There are many different topic model visualizations used in existing applications. These visualizations can be distinguished in different ways. For example, some visualizations show entire topic models, while others focus on individual topics. Then, visualizations can be static or interactive. Some model outputs are limited to textual representations, like displaying keywords in topics, while others use visual representations in the form of bubble diagrams or node-edge networks. Finally, we can make the distinction between explanatory and exploratory visualizations. Explanatory visualizations can be used to validate assumptions, while exploratory visualizations let the user explore and discover new insights. In addition to differences in visualization purposes, the type of data and how different types of data are visualized may vary. For example the number of words, but also way of ranking them, influences how topics are interpreted by humans. Some applications even show automatically generated summaries of topics.

Multiple papers introduce topic modeling result presentation based on word lists

[18], [9], [61] and [36]. Topic Browser [18] uses simple horizontal **word lists** for displaying topics, where each topic is represented by the two most occurring words in this topic (see Figure 6). The authors of this paper argue that showing the top 10 words in a cluster does not provide sufficient information to inform the user about the basic idea of what the topic captures, and therefore presents a wordcloud of the top 100 most occurring words of a selected topic in the interface (Figure 7). The **size of words** are determined by the word's probability in that topic. In addition, the relation of the top 10 words per cluster can be explored through word lists consisting of words in the same context. This word display is according to the authors most useful, because topics in a model cannot be fully interpreted when they are completely separated from their context. Topic Browser also enables document, word and attribute browsing through visualizations, all in the format of **lists**. All visualizations are **interactive** through sorting and filtering features. Document-level visualization in Topic Browser includes topic distribution information and presents similar documents based on that distribution.

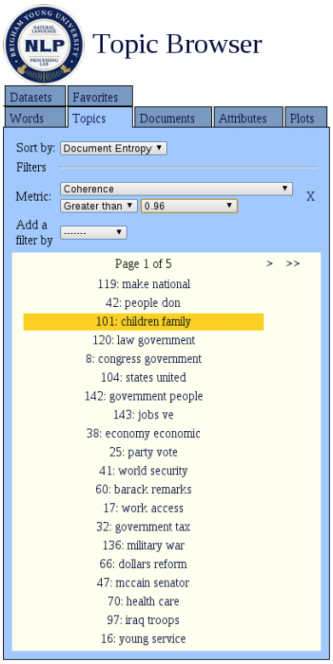

Figure 6: Topic Browser: Topics represented [18] by the two most occurring words per topic in a list.

Word lists representing topics or topic-document structures are also used in other previous topic models. Chaney and Blei [9] enable exploration on topic or document level in their Topic Model Visualization Engine. LDAAnalyzer, created by Zou and Hou [61], is another topic model visualization application that displays topic-document structures using **word lists**. Murdock and Allen [36] introduced Topic Explorer, which uses lists of words to present topic distributions within articles. Instead of plain horizontal word lists, documents are displayed in **bar charts**, which show the topic distribution per document with colors referring to

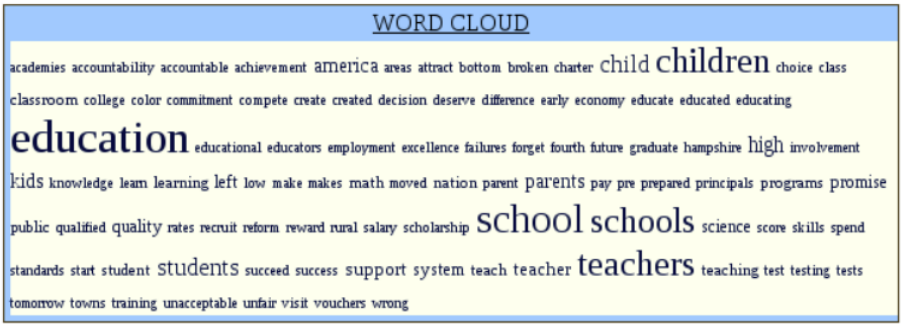

Figure 7: Topic Browser: Terms in individual topic [18], alphabetically ordered. The size indicates the probability of that word occurring in the topic.

different topics (Figure 8). This visualization requires more cognitive effort by the user compared to displaying word lists enabling fast evaluation by eye-balling. Bar charts, however, provide more information about topics and relations which might be beneficial for the user. This visualization method will be explored in the current setting as well.

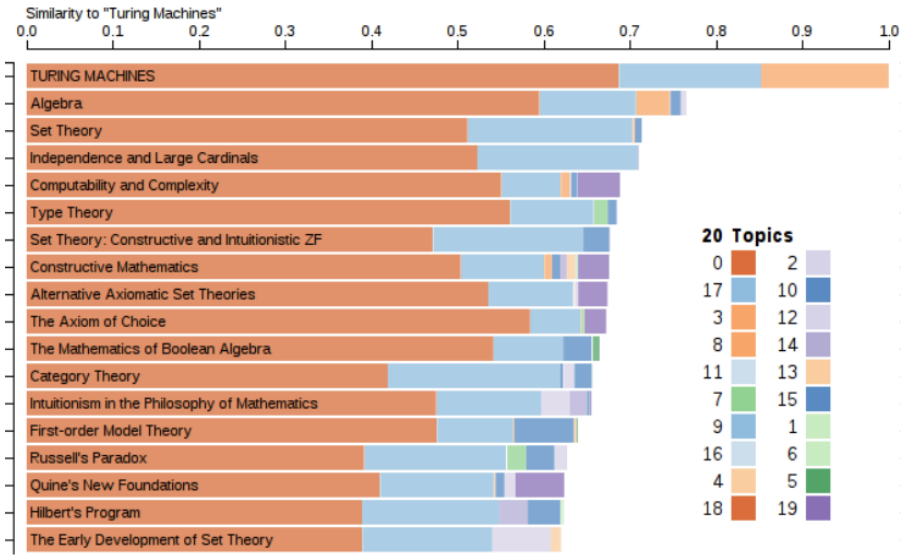

Figure 8: Topic Explorer: Individual articles (rows), its distribution and weights over topics (size and color of bands) and its similarity to the article 'Turing Machines' [36].

An alternative way of visualizing the model output is by **bubble charts**. Bubble charts provide additional information compared to the list methods described above, like relative topic size and relationships between topics. LDAVis, developed by Sievert and Shirley [48], is an interactive visualization with topics represented as bubbles (Figure 9). The topic bubbles are plotted in a two-dimensional space, displaying both the prevalence of a topic as well as its relation to other topics. Bubble centers in the displayed space are computed by scaling down high-dimensional topic distances to two dimensions. The **prevalence** of a topic is visualized by its bubble size. In

addition to the topic bubbles, the meaning of topics are revealed by a word ranking list on the right side of the view (Figure 9). Overlaid bars represent the word's frequency in the corpus and the topic. These words, although represented in a list, are ranked slightly different than other list-like visualizations. Keywords are ordered based on their **relevance**, instead of probability. This relevance is based on the probability of a keyword in a topic as output of the model, and the ratio of this probability to its marginal probability across the corpus. The effectiveness of showing words based on this relevancy ranking measure is, however, as far as we know, never evaluated in a user study. The visualization by Sievert and Shirley [48] is the only visualization that is open sourced, and will be considered in this research.

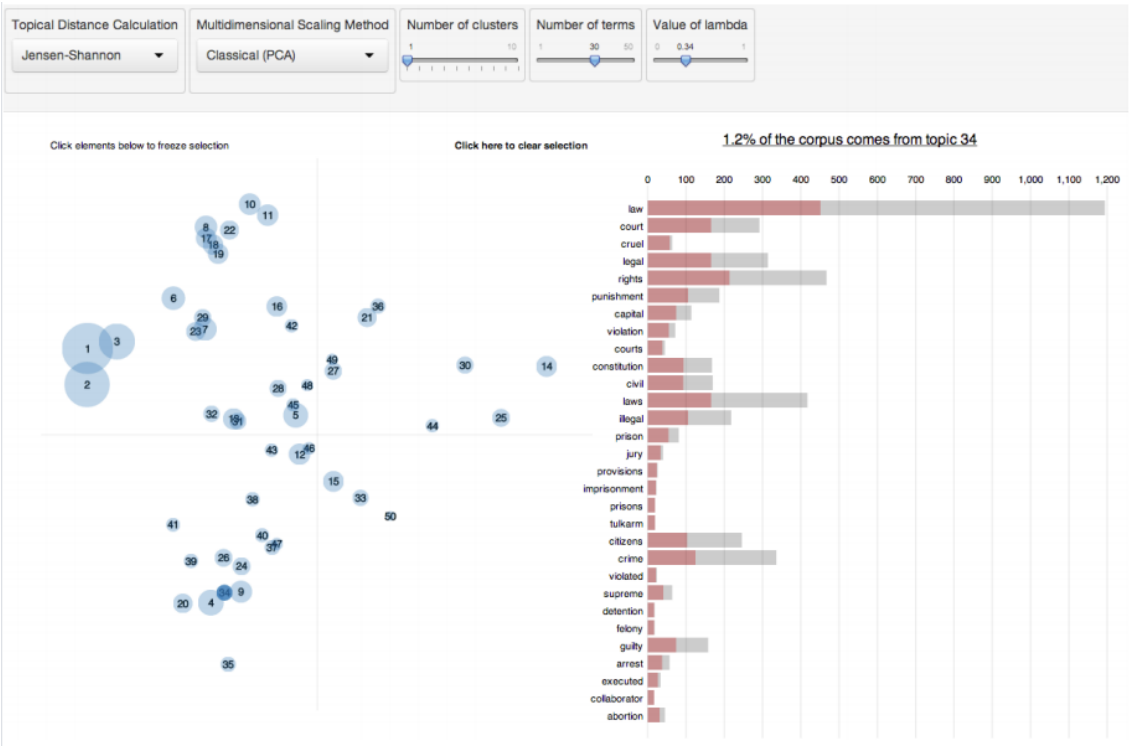

Figure 9: LDAVis: Bubble chart of topics, their relative size, position towards other topics and most prevalent words. [48].

Termite [12] (Figure 10) also uses bubbles to indicate relative importance, but then of single words in relation to other words in all topics. This is presented in a **grid view**, in which terms are presented against topics, to compare keywords in different topics. This enables users to discover significant words and coherent topics in the data. This visualization is useful for discovering relationships between specific set of words and the generated topic, and should only be implemented if the user needs to learn about the relative importance of words between topics. The current setting of interactive topic modeling on high level does not require such a detailed visualization on the relevance and relations between individual words and topics, and will thus not be considered further.

Furthermore, **network graph** structures are used in some studies to display

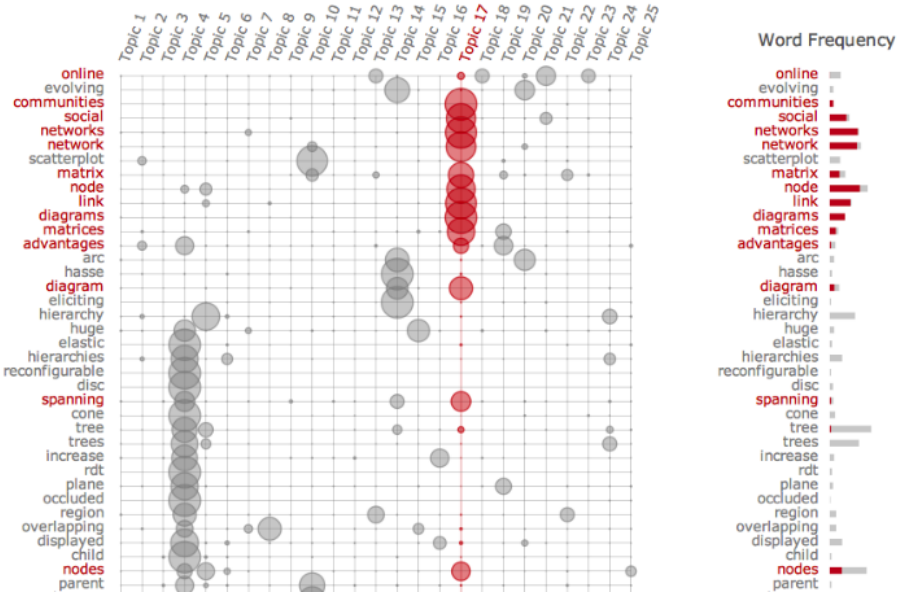

Figure 10: TERMITE: Grid view of terms in topics [12].

topics and its relations to each other [53]. Smith et al. [53] display contextual information in a network of topic nodes and word nodes, by computing term co-occurrence and topic co-variance in the model. Topic nodes and word nodes within topics cluster together based on their relatedness, using treemaps. Other tools provide even more information on the extracted topics. For example, Smith et al. [51] introduced TOPICFLOW, which displays **temporal changes** of the model using a Sankey diagram. Treemaps and Sankey diagrams, visualizing complex relations and temporal changes, surpass the high-level topic modeling goals of this research, and will not be considered in the current implementation.

There are different methods of sorting or ranking the words that represent topics. Keywords in topics are usually ranked based on **probability of occurrence**. This means that topics with a lot of documents belonging to this topic, and keywords that occur most often in a topic, will end up higher in both listed representations. This approach is applied in other visualizations described above that use word lists as well [9], [36] [61]. **Keyword relevance** is used in LDAVIS [48], and **saliency** in TERMITE [12]. **Saliency** ranks and filters keywords to select the most relevant terms instead of generic ones. According to the authors Chuang, Manning and Heer [12], surfacing discriminative terms lead to faster assessment and topic comparison by users. This is however, to the best of our knowledge, never evaluated in a user study.

Then, topics can be sorted in various ways as well. While most applications present topic lists ranked by **size** (e.g. the number of documents in the corpus in which the topic occurs), TOPIC BROWSER [18] ranks extracted topics based on the **coherence** of top ranked words. The more semantically similar the words representing a topic, the higher up in the topic list this topic will appear. High ranked topics are more likely to intuitively make sense to a human and less likely to

be an artifact of statistical inference.

2.2.2 User evaluation studies to topic model visualization

Smith et al. [54] studied and evaluated how users receive four different visualization techniques: word lists, word lists with bars, word clouds and network graphs. Users were asked to compare the four visualization techniques against each other with labels generated by users themselves and against labels that were automatically generated. Label quality was used as measure for how accurate users interpret topic from a particular visualization. Although no meaningful differences were discovered between the label quality of the four different visualization techniques, a difference between simple and complex visualizations was observed. Simple visualizations, like word lists and word clouds, support a quick initial understanding of topics, while more complex visualizations, like network graphs, take longer to understand but reveal relationships. Concluded is that there is no 'best' visualization technique in general. For efficiency, simple word lists are the best. Multi-words expressions and relations between words and topics may be obscured by simple visualizations, and thus more complex visualizations, like network graphs, are recommended.

Regarding topic labeling, Smith et al. [54] suggest to use a high word cardinality. When more words are used to represent a topic, it is less likely that a topic will be misinterpreted. Furthermore, they found that topics with a higher coherence are easier to interpret. Label analysis revealed that 'good' labels (evaluated by users), are, on average, shorter than 'bad' labels. In addition, users prefer topic labels containing general, descriptive terms, instead of words from the topic itself [37].

2.3 Interactive Topic Modeling (putting humans in the loop)

So far, topic modeling techniques and visualization techniques have been presented. With Topic Modeling (TM), it is possible to explore large amounts of text data by automatic topic extraction. More recently, Interactive Topic Modeling (ITM) (also known as Human-in-the-Loop Topic Modeling, HL-TM) has been introduced to take advantage of the domain knowledge of the user into the generated topic model. In contrast to static topic modeling, ITM allows users to influence the modeling process.

ITM has been applied in various domains, where applications are designed specifically for domain experts to interact with the model. First applications exist for information retrieval [59], but also outside the natural language domain like computer vision [17] and bioinformatics [33]. Other applications, introduced in previous studies, are developed for general purpose, to get insights from big document corpora [30] [11] [23] [52]. All these studies share the goal of incorporating domain expert feedback into topic models, to let them understand and explore themes in big data sets.

Some studies evaluate (parts of) new or existing frameworks with qualitative user studies or quantitative model performance measures, in order to gain insights on best practices regarding user experience and model implementation [2] [5] [31] [58] [50].

2.3.1 Design challenges in Human-Machine Collaboration

An ITM application involves human-machine collaboration, which has certain minimal requirements to ensure effectiveness. Here, a number of machine requirements are discussed that should be taken into account when designing an ITM.

First of all, effective collaboration requires *transparency*. If models are understood better, model mistakes can also be corrected better by users [28]. Second, the model should use the user's feedback to their expectations, to ensure *predictability*. Unfortunately, there is often a trade-off observed between transparency and predictability: high transparency, where controls are easy to validate, expects predictable outcomes and leads to difficulty in providing users with controls [50]. Models thus need to balance respecting user inputs and truly modeling the data. From human-computer interaction studies we also know that interactive interfaces and models should be *transparent*, *predictable*, *controllable*, and should provide *fast, continuous updates*, in order to be effective, efficient and trustworthy [1].

A strong relationship between *interpretability* and *trust* was found by Bakharia et al. [5]. To achieve high trust of the user in the topic modeling system, topic modeling results should be easily interpretable, for example by showing simple visualizations.

Smith et al. [52] did an extensive user study to the user experience of interactive topic models and challenges regarding machine learning like unpredictability, trust and the lack of control. This qualitative study, focused on *control* and *stability* of user refinements of the model, revealed that users prefer simplicity. Furthermore, they present design principles for future ITM applications. To achieve high user experience, ITM applications should: provide a history of actions and model results, support 'undo', have a saving option with reminders to save, allow topic freezing and support multi-word refinements.

In a follow-up study with an iteration on the used ITM test application design, Smith et al. [50] concluded from user studies to ITM applications with distinct approaches (variations in *model adherence*, *stability*, *latency* and *quality*) that users dislike *latency* the most. A lack of *adherence*, whether the user's input is applied as expected, came out as second most prevalent dislike. In addition, they found that users want to be heard. User input should be reflected in the model output, and unexpected changes or changes that cannot be fully incorporated should be explained. Users are willing to share control with the system, but only if the model informs the user continuously, and the users are able to undo changes and lock parts of the model. An interesting insight is that users had polarized opinions on model *stability*; some users like to see unexpected, new information on new runs, while others did not. They conclude with four recommendations: users want to be in control, users want speed, (unexpected) model output changes should be explained and parts of the model should be lockable.

The use case domain and user expertise impact how the model is perceived and used, hence should also be considered in the ITM design. Machine learning experts or advanced data scientist expect most likely have a rough understanding of how the model works, so they are able to assert model flaws, like unexpected results or instability. Domain experts without this background who have less understanding

and thus a different perception of the model, are more likely to become frustrated if the model is not adherent, stable or fast. Similarly, personality traits affect user experience. Thus, the user's background in expertise and personality should be considered in the design of the interactive model.

2.3.2 Existing frameworks

iVisClustering [30] (see Figure 11) uses LDA for clustering big document data, to allow users to have full control of the usually high cognitively demanding clustering task. The framework applies both bottom-up and top-down model revision approaches. In bottom-up clustering, the user starts with empty clusters and creates clusters with machine learning assistance. Top-down modeling, on the other hand, starts with topic model results and allows user to optimize this result via interactions. User revisions include deleting, merging, sub-clustering and word refinements, enabled by drag-and-drop interactions. Their framework consists of a lot of visualizations that allow the user to explore modeling results extensively (see Figure 11). A node-link graph is used to visualize clusters and their relations, and with text, colors, bars and links, details and relations topics, documents and words are visualized individually as well. In addition, they present a cluster tree view to display user interactions, a trace view show changes made by the user. These latter two visualizations are a unique contribution, since, to my knowledge, this was not applied in other applications. Although these different types of visualizations, especially the unique trace view, are promising, they have never been evaluated in a user study.

In 2013, Choo et al. [11] (see Figure 12) introduced the ITM application UTOPIAN, aimed to derive topics from real-world document corpora. In contrast to iVisClustering, UTOPIAN uses NMF as topic modeling method. More specifically, they introduced semi-supervised NMF (SS-NMF), in order to incorporate user feedback in the matrix factorization process. Quantitative experiments showed that this method outperformed the probabilistic approach LDA in terms of consistency and convergence time. A deterministic (high consistency) and low running time (empirical convergence) of SS-NMF are important factors to achieve high user experience. This framework contributes by introducing real time visualizations, revealing the modeling process before convergence, thus while the model is being updated. Clusters and their relations are visualized using node-link diagrams, where the dimensionality is reduced using the t-distributed Stochastic Neighbor Embedding (t-SNE) [34] method.

ConVisIT [22] is an application developed to interactively extract topics from asynchronous online conversations. The included users study shows that their ITM application was preferred over other methods of data analysis. The ConVisIT framework is unique in that it is specified to online conversations in web forum threats, and thus models rather small texts instead of general, long documents in other ITM frameworks. Nevertheless, this research contributes to the field in that it uses sentiment analysis in the modeling process. The application has a very rich interactive visualization platform, allowing the user to explore and revise extensively.

Saeidi et al. [45] developed ITMViz, which allows user revisions to the LDA

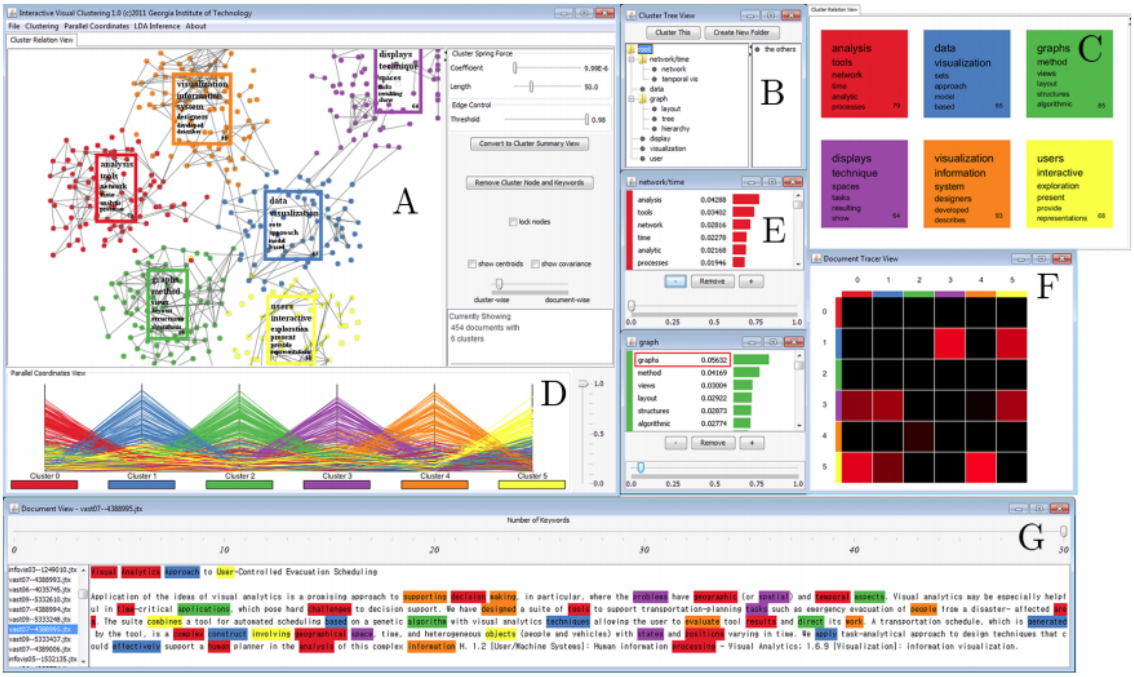

Figure 11: Overview of iVisClustering [30]. (A) Cluster relation view, visualizes topic relations. (B) Cluster tree view. (C) Cluster summary view, clusters simplified in words. (D) Parallel coordinates view, with the topic distribution of each document. (E) Term-weight view for each topic and modification options. (F) Document tracer view, which shows how documents change after user interactions. (G) Document view, with highlighted keywords that indicate topics.

model via must-link and cannot-link constraints. Although the limited revision possibilities, they showed that constraining topic models by domain knowledge contributes to extracting more meaningful topics. Unconstrained and constrained LDA model results where compared using the MoJo similarity measure, a clustering distance metric [56].

2.3.3 Revision techniques

The frameworks presented use various revision techniques. Summarized, these techniques allow users to make changes to the model on topic, keyword or document level. All these top-down revisions are summarized in Table 4.

2.3.4 Evaluation of existing ITM applications and techniques

Multiple other studies also show that interactive topic modeling is preferred over traditional topic modeling where the end-user domain expertise cannot be incorporated in the model. Hu et al. [23] showed that ITM significantly improved the topic quality by evaluating them with an assigned *variation of information* score, as well as a user study.

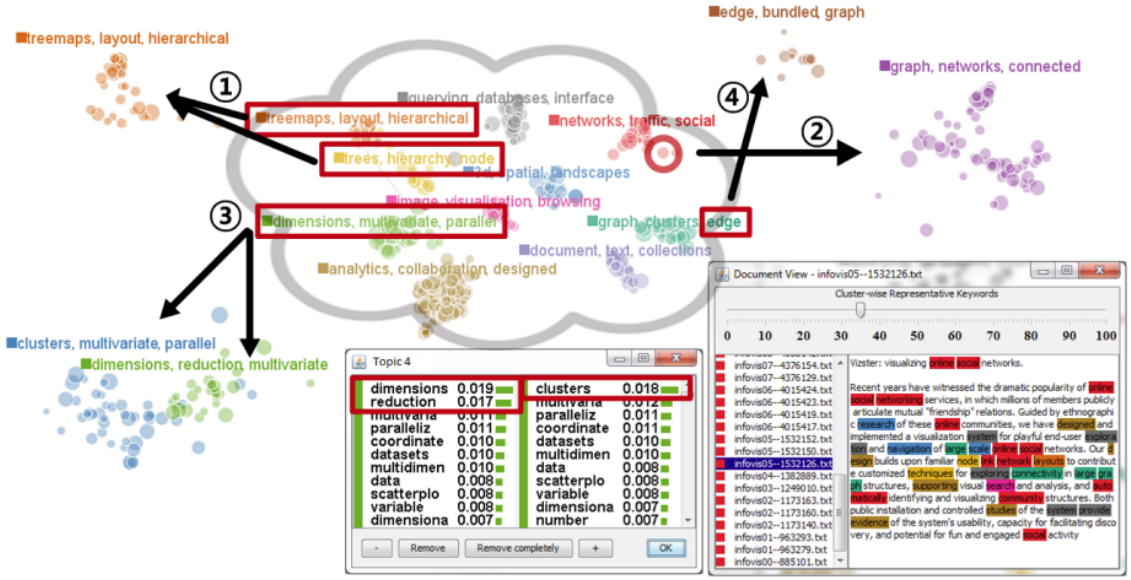

Figure 12: Overview of UTOPIAN [11]. A scatter plot of clustered documents generated by t-SNE. Revision interactions provided in this view are: topic merging (1), document-induced topic creation (2), topic splitting (3), keyword-induced topic creation (4), topic keyword weight refinement (left window) and keyword highlighting in documents (right window).

An extensive ITM user study was done by Lee et al. [31]. They performed open-ended interviews to participants to find out how humans understand, assess and refine topics. They used Wizard of Oz testing on their application which had many topic refinement options available in the user interface (Figure 13). Their main finding was that topics may be misinterpreted because of the words representing them. Based on words alone (no relations), topics may be hard to understand, especially topics that lack coherence. They recommend that topic refinement should be focused on topics with low coherence. Here, coherence was measured by normalized pointwise mutual information (NPMI [8]). In addition, they suggest to support refinement options to lower the cognitive load for the user, for example by offering suggestions. All refinement options used in the user study appeared to be useful, and they recommended including at least the most frequently used once in future ITM applications: add words, remove words, change word order, remove docs and split topic. Users prefer to have immediate feedback, and to ensure low latency they mention that refinements do not always have to be implemented in the back-end model.

Frameworks differ in TM technique, visualizations, and revision interactions. Various evaluation methods were applied, from technical TM assessments to qualitative user studies. Unfortunately, not all frameworks are evaluated, and because of the inconsistent evaluation methods, models can hardly be compared, it at all. What we can conclude, however, is that the context of the use case and background knowledge

Table 4: Revision techniques in existing frameworks

	Revision technique	UTOPIAN [11]	ConVisIT [22]	iVisClustering [30]	Hu et al. [23]	Termite [12]	ITMViz [45]
Topic level	Split topic	x	x	x			
	Merge topics	x	x	x			
	Remove topic			x			
	Label topic				x	x	
Keyword level	Create topic from selected keywords	x					
	Change keywords weights or order	x		x			
	Add keyword to topic				x	x	
	Remove keyword from topic			x	x	x	
	Remove keyword from all topics (stopword)				x	x	
Document level	Create topic from selected documents	x					
	Add document must-link constraint						x
	Add document cannot-link constraint						x
	Add document to a topic			x			
	Remove document from a topic			x			
	Remove document from all topics			x			

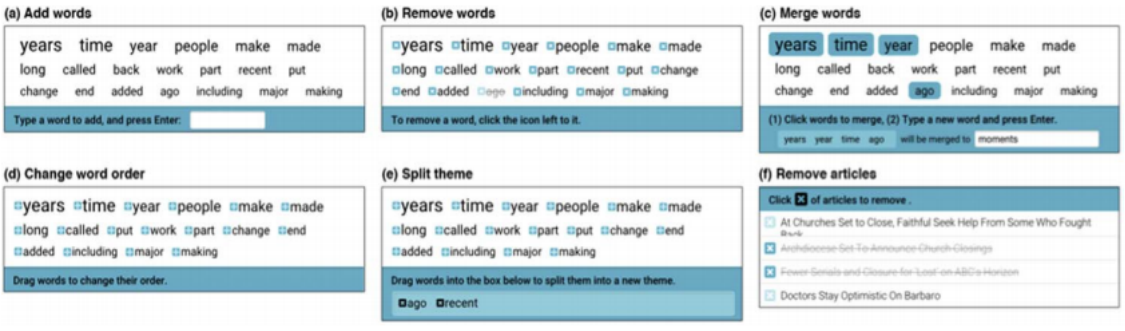

Figure 13: Overview of the possible user interactions in the user interface used in the user study by Lee et al. [31].

of the end-user determine what techniques, visualizations and revision options are best to use. Trends that were observed are the following. NMF and LDA are the

most frequently used topic modeling techniques, where NMF is more consistent and has a lower latency, which is beneficial for interaction. Then, model outputs should be exploratory and informative, but easy and fast to interpret. This can be achieved by offering simple views using words to describe topics, but offering additional visualizations to explore relationships between topics, documents and terms. Frameworks mostly apply top-down modeling, where the application suggests a topic model output and allows the user to refine this. To improve efficiency and effectiveness of user interactions, the application could offer suggestions where to apply model revisions. Information on the quality of individual topics, for example measured by coherence, helps the user in deciding whether revision should be applied. Revision possibilities vary a lot between applications, but user studies reflect that splitting and merging topics, and adding words, removing words, removing documents from topics are the most frequently used modifications.

2.4 Summary

The most common topic modeling methods are LSA, pLSA, LDA and NMF. pLSA and LDA are probabilistic approaches for deriving latent topics from text corpora. LDA has a generative component, which makes the method suitable for assigning topics to unseen documents. LSA and NMF use matrix factorization to reduce the dimensionality of the document-term matrix to identify latent structures in data. Results of LSA might be hard to interpret because they can be negative. This is overcome by NMF, which introduces a non-negativity constraint. NMF results in term-topic and document-term matrices that are easy to interpret and manipulate.

There are various ways of displaying topic modeling results. Most previous studies present results on topic, document and term level. While some present results only textually, most applications include visualizations which are sometimes interactive. Six different topic level representations can be distinguished:

- List
- Lists with bars
- Word clouds
- Bubble charts
- Grid layout
- Network graphs

Within these different representations, topics and words can are ranked in various ways, as described above. In summary, these are the most common methods used for topic and keyword ranking:

- Probability
- Relevance

- Saliency

Interactive Topic Modeling applications are introduced to take advantage of domain knowledge of humans. Previous ITM application designs differ fairly per use case, in terms of TM technique, visualizations, and revision interactions. LDA and NMF are the most used embedded topic models, visualizations are aimed at exploration and model revisions are mostly top-down.

Human factors need to be considered when designing an ITM application. Human-computer interactions should effective, efficient and trustworthy to achieve high user experience. To meet these requirements, most important challenges that have to be faced to achieve high user experience are model transparency, model adherence, predictability and model stability, controllability, low latency and output quality and interpretability.

To my knowledge, no framework exists in the domain of news and broadcasting companies. Besides this new domain, no framework has been presented which enables directly applying extracted topics for exploratory data analysis in bigger scope. The goal of this research is to present a novel data exploration method to assist domain experts in analyzing produced and consumed article, audio and video content. The aim is to fill the gap between the lacking domain knowledge of data science experts and lacking data science knowledge of domain experts. Presenting an interactive topic modeling application to domain experts, to extract latent themes from their media content, is part of this wider goal. TM results can be used in data analysis, combined with other data sources, to explore and gather new insights. Thus, the use case and context differs from previously presented applications. This new context affects design choices and possibilities. Since in the domain of news and broadcasting companies, modeling results can be used in combination with other data (e.g. consumption data with demographics), ITM application evaluation could be addressed from this new perspective as well.

3 System design and implementation

The previous section introduced related work on Topic Modeling and Interactive applications. This section presents the requirements of the ITM application that were gathered from users, and the design choices that are made upon them. The requirements serve as basis for the next step in the human-centered design process: the solution design, development and implementation, which are also presented in this section.

3.1 Requirement Analysis

The second step of the HCD process (Figure 1) is gathering and specifying the user requirements. Initial requirements are gathered in the first design cycle using an in-depth user study, but are refined in next iterations by evaluations of the design against these requirements. This subsection presents how and when they were gathered. The study protocols and results are included in Appendices B.1 and C.

First, different types of users and other stakeholders are identified by desk research, and classified as primary, secondary or tertiary users. The identified stakeholder groups are included in Appendix A. Primary users are the potential end-users of the application, domain experts within the broadcasting company like data-literate journalists and media content analysts. Secondary users are journalists, content producers and media planners who are interested in the outcome of media topic modeling in their decision making. Tertiary users are managers, system administrators and product owners, who are not considered any further in this research.

Then, user research was done prior to developing the application, to gather user and business requirements and constraints. This formative user study was conducted to potential end-users and secondary users. The user research applied here to gather user requirements is more attitudinal than behavioral; we want to discover what people want to achieve with the application before the product is developed. In addition, the user research is qualitative rather than quantitative; we want to gather information from the users *directly*, aiming to answer questions about *why* and *how* to fix a problem. Within the context of the thesis and the company Yle where the ITM application has been developed, the initial scope was kept small. The initial stakeholder and user group is small, so that small proof of concepts can be developed without requiring large resources. This implies that the available user group for this research and participatory design is also small. In the initial user research to requirements engineering, no product or topic modeling application has been used. Later, participatory design methods were applied, where the users interact with (parts of) the product prototype for evaluation and design purposes.

Following user recommendations of Rohrer [44], keeping the goals and constraints described above in mind, the first user research to understand the user's desires and gather initial user requirements, was in the form of a **focus group** session. This form of user research is a more managed process with specific participants than an informal brainstorming session, but more flexible and more open than individual interviews. The main two components of the initial focus group session were (1)

learning about the general user background and attitude, and (2) doing a Wants and Needs (W&N) analysis: a fast brainstorming method to gather requirements from multiple users simultaneously [6]. Three potential stakeholders (two potential end-users and one secondary user) where led through a semi-structured discussion about the product idea. The study protocol and results are included in Appendix B.1 and Appendix C respectively.

Following this session, recurring one-hour sessions with two focus groups were organised (in-person as well as remote). Sessions with these groups were used in the design, development and evaluation process using a combination of three user study techniques: **Participatory design**, **Concept testing** and **Desirability studies**. The three techniques are attitudinal, mostly qualitative, and all incorporate hybrid product or prototype usage during data collection [44]. The purpose of these sessions was multifold: (1) evaluation of proposed model and design solutions (from step 3 of the HCD process) against the requirements (step 4 of the HCD process) through **concept testing**, (2) proposing and evaluating new (model, design, visualization) elements by **desirability study** methods (step 3 of a new HCD iteration) and (3) learning about what model and design elements matter most to the users and why using **participatory design** methods (step 2 and 3 of a new HCD iteration). The study design and results are discussed in detail in Appendix B.2.

3.2 Design choices

The user requirements identified in the previous subsection specify *what* the application should do, not *how* it should be done. In this subsection, the engineering specifications derived from user requirements are combined with findings from related work. This results in a set of design choices for the ITM application.

3.2.1 Topic Modeling Technique

Multiple topic modeling techniques were discussed in Section 2.1. The different methods and their characteristics are summarized in Table 5. The techniques are briefly revisited, along with the user requirements, to motivate choice of topic model to implement in the ITM application.

Requirement RM3 (see Appendix C) states that topics in documents must be represented as distributions. Since this is a 'must have' requirement, K-means clustering will not be further considered as possible modeling method. Then, requirement RM10 states that the topic model must be able to assign topic distributions to new documents, outside the training corpus. pLSA does not assign parameters to model the probability of a document (P(d), see Section 2.1.3), it is not possible to embed new documents. pLSA is a generative model of the trained documents set, it is not a generative model of new documents. pLSA thus violates this requirement, and is not used as modeling technique. The other five listed requirements are 'should have' requirements. As mentioned in Section 2.1.5, NMF is a similar method to LSA, but with a negativity constraint. This results in higher model interpretability and higher model output quality, as reflected in the comparison table. Thus, NMF is preferred

over LSA.

Then, LDA and NMF, which are also the most used methods in previous ITM applications, differ in strong and weak points. One way of comparing topic model interpretability is by measuring the intrinsic topic coherence. As mentioned in the background (Section 2), a high topic coherence indicates that the topics' top terms (the most frequent words in a topic, that are usually presented to the user) are semantically interpretable. A high topic interpretability is preferred when the user is judging on the topics by just 'eye-balling', for example to make fast decisions on model refinement like selection the optimal number of k. O'Callaghan et al. [41] found that NMF produced more coherent topics, while LDA produced topics with more generality and redundancy, measured by different coherence measures. They suggest to use NMF in favor of LDA, especially if the corpus is associated with niche or non-mainstream domains. In addition, NMF has better model consistency, lower empirical convergence time and user feedback is easier to incorporate in its non-probabilistic, matrix decomposition model. NMF and LDA are compared in detail in Appendix D.1. Since better model consistency and lower empirical convergence (and running) time are aspects that are found desirable (in the user research and related work), NMF is the preferred method to apply in the ITM application in this thesis.

Table 5: Topic Modeling method comparison against requirements.

Requirement	**K-means clustering**	**LSA**	**pLSA**	**LDA**	**NMF**
Topic-Document distribution (RM3)	- (single)	+ (multiple)	+	+	+
Possibility to embed new documents (RM10)	+	+	-	+	+
Topic Model quality	+ (high)	+/-[3]	+[3]	+	+
Model output consistency	+/-[4]	+ (high)	- (low)	-	+
Model empirical convergence time	- (high)	+ (low)	-	-	+
Model output interpretability	+ (high)	- (low)	-	+	+
Difficulty of incorporating user feedback	+ (easy)	+	- (hard)	-	+

[3]Depends on the size of the input dataset, large datasets generally leads to better topic quality.
[4]Depends on settings

3.2.2 Data preprocessing and feature extraction

NMF decomposes a document-term matrix into document-topic and topic-term matrices, consisting of weights and features respectively. The terms in the original document-term matrix are thus the features the model is based on. These terms are usually not all the terms that appear in a raw text document. Applying NMF requires preprocessing the set of documents. Without preprocessing, thus taking raw document texts, modeling leads to meaningless and incoherent topics. This is because the documents will be over-represented by common words (*stopwords*) that occur in natural language. In addition, words can appear in many morphological variations, and without lemmatization (or stemming), the model will not recognize the same words in different morphological variations as one. The following preprocessing steps will be applied to the raw document texts:

1. Constructing a bag-of-words, by:
   (a) Removing capitalization and punctuation. All words are reduced to lowercase for simplicity.
   (b) POS tagging (Part-of-Speech tagging) and keep only nouns, verbs and adjectives. Other parts of speech are carry insufficient information for topic modeling.
   (c) Lemmatization, to remove inflection.
   (d) Stopword removal. Removing common terms that are connecting parts of a sentence rather than showing subjects, object or intents.
2. Feature extraction:
   (a) Normalizing each document-term pair by the tf-idf weighting mechanism.

The bag-of-words (BOW) model is a simplified representation of text in a corpus. Documents are represented as a multiset (bag) of the words it contains. The BOW model first builds a vocabulary of all the unique words present in the whole corpus. Then, a sparse vector (with the length of all the words in the vocabulary) for each document is constructed, which is filled with the number of times the corresponding words occurs in a documents. BOW thus disregards grammar and word order, but keeps its multiplicity.

Although BOW can be used as feature for training a topic model, NMF performs better when the documents' bag-of-words are normalized over the corpus. *Term frequency-inverse document frequency* (tf-idf) is the most frequently used feature extraction method for text data, and NMF. Appendix D.2 explains how tf-idf is constructed for a set of documents.

3.2.3 User Interface and Interactions

The user interface of the ITM application will be the place where the users interact with the topic model. Users will have to select data, configure the model, refine the

model based on its initial output, and perform exploratory data analysis on the final topic output. Complex model settings and computations will be hidden from the user, by embedding it in the background of the application. According to the user research and technical requirements, the user interface will be designed according to the following five steps that are cohere to the mental model of a non-technical end-user:

1. **Data input selection**. Users want to select the data source, filter by metadata and choose the time window. Immediate feedback to the user's input should be presented in the application, but this does not have to be a graphical visualization; users indicate that a table consisting of all data would suffice. Additionally, some data quantity and quality measures will be presented, to prevent unexpected modeling results as result of bad data input. Additional user requirements and how they affect the user interface design are discussed in Appendix D.3.

2. **Model configuration**. This step consists of (hyper)parameter setting and optimization by the user *before* extracting topics. The only hyperparameter of the topic model itself is the number of topics k to extract. From literature and focus group sessions we know that choosing the optimal number of topics before an initial model run is challenging. Choosing too few topics may lead to very broad results, while too many topics may produce may small and similar topics. Although the user will be able to change k and rerun the model at any time, a topic suggestion function will be built in the application. Topic coherence of 5-25 topics will be computed based on a Word2Vec model, and results will be graphically presented. In addition, the user will be given the option to tune the feature selection of the training data. The document set is already preprocessed, but the user will be given the option to filter out extra stopwords and change the minimum and maximum *document frequency* (df) of the tf-idf selection. End-users in the broadcasting domain suggested that they prefer to be in control which stopwords to exclude over automatically excluding a predefined set of words, because words to exclude might be use case specific. Visual representations of frequent terms in the documents should help the user to decide on words to manually exclude and how to change the df values. Justification for the topic coherence measure and details on the hyperparameter tuning of the tf-idf feature selection are presented in Appendix D.4.

3. **Model output interpretation and evaluation**. This step is important for the user, because it informs about which topics to refine in the next step. Three model evaluation perspectives will be included in this application component. First, the model output will be visualized from a topic-term point of view. Users should be able to get a quick overview of the generated topics, by assessing which words are represented in the topics and how the topics relate to each other. Eyeballing over the topics is possible by presenting the topic-term table with most prevalent words per topic. The relation between topics and its words will be presented by a visualization similar to LDAVis (Figure 9). Second, users

will be able to learn what topics are reflected in the input set of documents, enabling them to evaluate whether the extracted topics make sense to their perspective. The document-topic matrix will give the users a quick overview, but relationships between documents and topics can again be best visualized in a two-dimensional plot. Users indicated interest in comparing individual documents, so an interactive stacked bar chart showing the distribution of topics represented in selected documents will be the third visualization from document-topic perspective. Finally, users indicated that they would like to see suggestions on what topics need refinement. Although topics could be best evaluated by human judgement, quality of individual topics can be estimated by model residuals. Average residuals will be computed and presented to the user in the application. Elaboration on and justification of all visualizations and its methods is included in Appendix D.5.

4. **Model refinement**. This is the step where the human can manipulate the topics of the model output, based on subjective evaluation supported by the visualizations. Iterations with focus group sessions resulted in a final set of options to **merge**, **split**, **remove keywords from a topic** and **rename a topic** (*manual topic labeling*). The refinement options are simple operations; users prefer simple interactions over changing complex details (e.g. changing weights of single keywords in topics). The results of the refinements will be directly reflected in the visualizations. Appendix D.6 presents why these specific refinement are chosen, reasoned from focus group sessions and design iterations.

5. **Exploratory data analysis of the output in wider context**. This step is usually not included in ITM applications, but in this context the analysis of the model output is considered as equally important to the other steps. The purpose of the application, which was learned from the focus group sessions, is to extract topics from various (subsets of) data that can be used in broader perspective to explore and gain new insights. In collaboration with the stakeholders in the focus group sessions, two forms of exploratory data analysis (EDA) will be included in the application: topic development over time and comparing topic content production with consumption by different age groups. More details on what data these figures visualize and how they are designed are included in Appendix D.7. This section also includes how data export is supported in the application. All data and figures used and generated in the modeling process, as well as all the interactions of the user during the modeling session, can be exported.

Overall, in addition to dividing the ITM application into the five steps described above, the application's interface has, like many interactive interfaces, two main functions: enabling user input and providing information (feedback on user or system actions). In order to make the application as intuitive as possible, the interface will consist of two main components, horizontally separated, each meant for one function. The five-step process will be vertically present in both components, guiding the user through the process naturally down the page. So, all information and feedback

(visualizations, model output) will be presented in the main component of the UI window, while the user input will be possible in a smaller panel on the left side of it.

Moreover, *Jakob Nielsen's 10 general principles for interaction design* [38] are followed as much as possible in the design of the ITM application. How these 10 usability heuristics are applied in the application design is included in Appendix D.8.

3.3 Implementation

3.3.1 Front-end

The previous subsection introduced five steps the user interface is divided in. This design is clearly reflected in the implementation to the user as well, since every step requires different user actions. The user flow is presented in Figure E3. This diagram describes the user journey of using the application for interactive topic modeling, from data selection to exploratory data analysis and exporting results. The five steps of the application are also reflected here, but the user is free to jump back to any earlier step, for example to change the input or model settings. This diagram only reflects the user actions in the application; actions from the topic model and back-end architecture are visualized in separate diagrams (see Figures E4 and E2).

The front-end of the application is built with the open-source application framework *Streamlit*[5]. *Streamlit* is a tool to build interactive web applications around data and machine learning models, with *Python*. Like Jupyter Notebooks, Python scripts for data modeling are the only scripts required to construct a user interface; the Streamlit framework builds an interactive and user friendly interface around it. In contrast to Jupyter Notebooks, which is a web-based interactive computing program, Streamlit builds web applications using its own HTML, CSS and JavaScript, by providing a Python API. Where Jupyter notebook requires users to run blocks of Python code, Streamlit hides the script in the background and presents an interactive interface to the user. An alternative that was considered for the ITM web application, with a Python model in the back-end, was the *Flask*[6] framework. Deploying Flask apps requires writing HTML, CSS and JavaScript, to create a user interface on top of the data model in Python. Although Flask allows more design freedom than *Streamlit*, it requires a longer design, development and deploy process. Streamlit offers high performance while maintaining the flexibility of rapid prototyping. Using Streamlit's API was thus preferred for the iterative design and development process that was maintained in this project.

Figure E1 shows Streamlit's workflow. Every time the user interacts with the interface, the Streamlit Python script is run from top to bottom to update the UI. To make the application performant, only a subset of the pipeline is recomputed to display the change in the user interface, while the rest is taken from cache.

All but two visualizations are directly shown when the user interface is rendered. Two interactive visualizations, presenting relationships between topics and documents in

[5] https://www.streamlit.io/
[6] https://flask.palletsprojects.com/en/1.1.x/

a two-dimensional space, are only generated on explicit user request. Generating these two visualizations requires a dimensionality reduction computation, which is kept outside of the default UI rendering process for performance reasons. All visualizations, except for these two figures, are made using Plotly[7], a Python open source graphing library. In contrast to the widely-used Matplotlib[8] library, Plotly is browser-aware. Instead of creating non-interactive charts that Streamlit would display as static images, Plotly does not require Streamlit to re-render the entire interface for a interaction with a figure. For performance reasons, all visualizations are created using Plotly, which sends interactive charts to the browser.

Although Streamlit (and additional visualization libraries), comes with a clean and user friendly interface design, some customization of the interface was performed. First, the default size of the left sidebar is quite narrow. Because of the design choice to put all user input in this sidebar, this sidebar is widened by manipulating the CSS file. In addition, the main color in the figures is set to turquoise (#00b4c8), which is the main color of the brand Yle[9]. Although this was not a direct requirement by the stakeholders, the organization requires to use the Yle colour palette in corporate-level publications, presentations and graphics. End-users of the ITM applications might use PNG exports of the figures generated in the ITM application directly in data reports and presentations representing the company, so Yle's colour palette is used as much as possible following this indirect requirement.

UI Components

As mentioned in the previous section, the user interface is horizontally divided in two elements. A main panel displays information in the form of text, tables and (interactive) visualizations. A panel on the left takes in most user actions, which influence the model. Changes made in the left panel are always immediately reflected as output or changes in existing visualizations in the main panel. The main panel allows some user input, but only for data output interaction (toggling visualizations and interacting with visualizations), and does not influence the model.

The entire interactive topic modeling application consists of a only single web page. A single web page provides a clear overview of the whole application, where all interactions and data visualizations are easy to review by the user at any time. However, not all components are directly displayed at the application initialization. The page extends during the user journey, to guide the user through the modeling process and preventing missing an essential step.

When the user enters the web application, only steps one (data input selection, Figure 14) and two (model configuration, Figure 15) of the modeling process are presented. This allows the user to focus on these steps, which essential for modeling good topics. The user is given the option to open a detailed how-to-use explanation. Here, the five main elements and required user actions of the application are explained.

[7] https://plotly.com/
[8] https://matplotlib.org/
[9] https://yle.fi/aihe/artikkeli/2015/10/22/yle-brand-visual-image-and-logoscolours

In addition, the working of the topic model itself is briefly explained, which helps the user in understanding the influence of data input and parameter tuning on the output. Tips are given to support the user in obtaining good modeling results and how to interpret the visualization to utilize them in refining the model. Contact details for further questions or issues are provided.

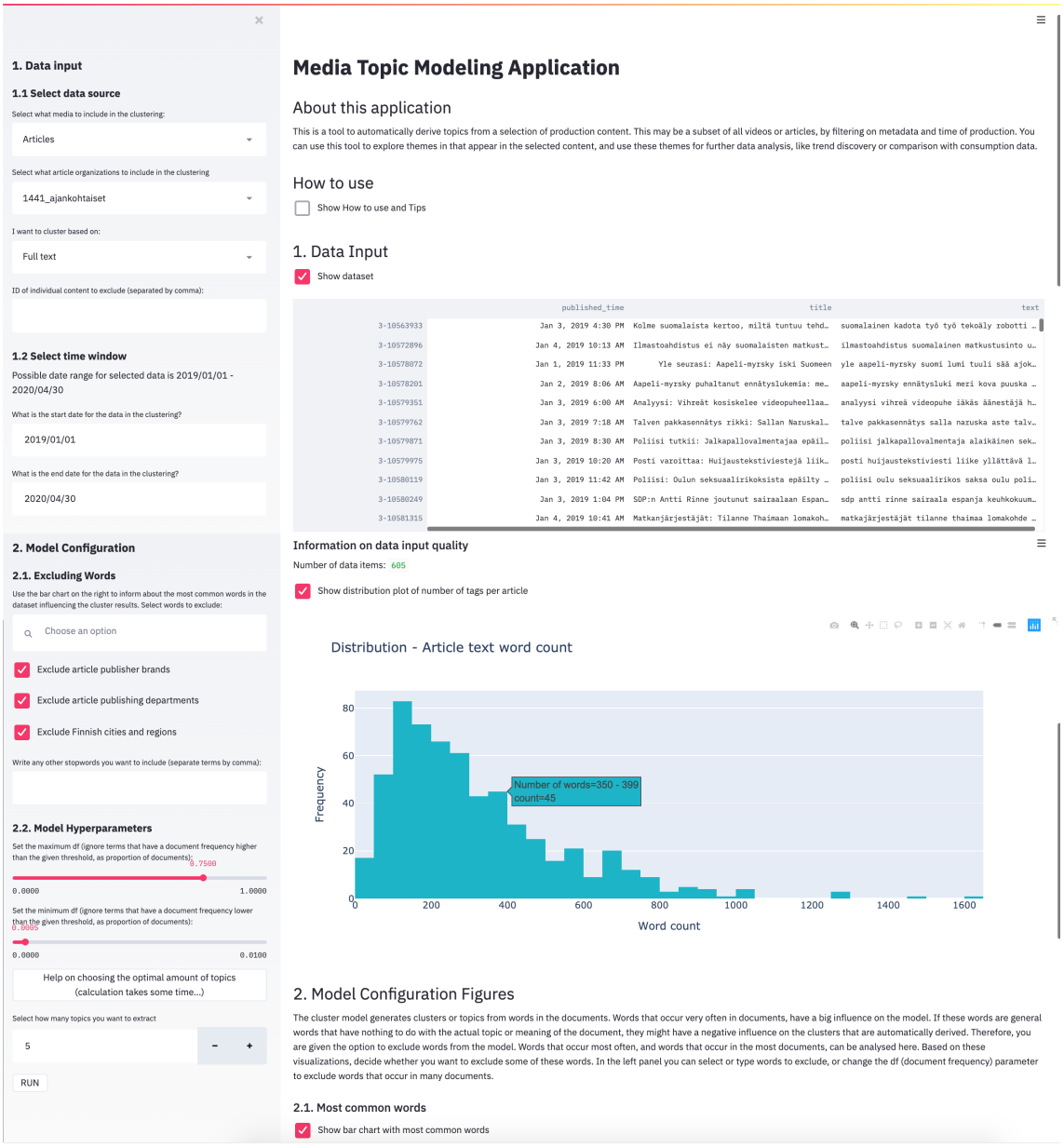

Figure 14: Screenshot of the landing page of the ITM web application. User step 1 (data input selection) is displayed. The left panel is for user input, the main panel visualizes the output of the actions on the data and model in real time. As an example, the articles published by the department of Current Affairs from January 2019 until April 2020 are selected. A preview of the preprocessed data and distribution of word counts in the article texts are visualized.

Then, the main panel shows the data input. The actual data input is presented in a (scrollable) table. Tables are made interactive: users are able to sort the table per column by clicking on the column's name. The number of documents is shown, as well as a bar chart of the distribution of the number of words in the documents, to support the user in making an estimate on the data quality.

The left panel includes drop-down menus for data source selection and filtering on metadata. The time window can be selected via an intuitive calendar selection widget, which pops up when selecting a time input box.

The second step, model configuration (Figure 15), includes a bar chart of the 15 most prevalent words in the document corpus selected in the previous step, as well as a table showing the document frequency of each word, sorted by df value. While the former visualization supports the user in manual stop words selection, the latter gives information on the df parameter setting. The left panel allows manipulating the feature extraction by excluding words from the model, either by selecting a popular word using a drop down menu with the most common words, typing additional words, clicking checkboxes to exclude predefined sets of words or changing the minimum or maximum document frequency using sliders. When the user requests help on choosing the optimal amount of topics for the selected dataset by clicking on the button in the left panel, a line chart of the topic coherence (y-axis) per k (x-axis) will be shown in in the main panel after calculation. Next, the user can select the number of topics to extract, and press the button to run the initial model to extract the k topics.

As soon as the topic model is initialized and the topics are derived, the remaining part of the interface (steps 3, 4 and 5) is rendered to allow output interpretation, data analysis, model refinement and data exportation.

Section 3 of the main panel displays a number of tables and graphs (see Figure 16), as explained in the previous section (D.5). Checkboxes are implemented to toggle tables and figures, and buttons are shown to generate additional, more computational demanding, interactive visualizations.

The interactive topic visualization to explore relationships between topics and keywords is only generated on user request (Figure 17). This visualization is made using a Python library for LDAvis (pyLDAvis[10]), which generates an interactive HTML file from the output of a topic model. Although this topic model visualization method is designed to visualize results from LDA topic modeling, it can also be used to visualize NMF topic model results, since it takes the generated topics and most relevant keywords as input which is also provided by NMF modeling. The topics are projected on a two-dimensional plane using Principal Component Analysis (PCA) on a distance matrix created using the Jensen-Shannon divergence on topic-term distributions. This multidimensional scaling function was preferred over other methods that pyLDAvis provides, like t-SNE, because it is faster and better visualizes the relations between topics on the two axis according to the semantics (UMAP, as discussed in Appendix D.5, might preserve the relative distances between the topics even more on a high performance, but this method is not provided in the pyLDAvis

[10] https://github.com/bmabey/pyLDAvis

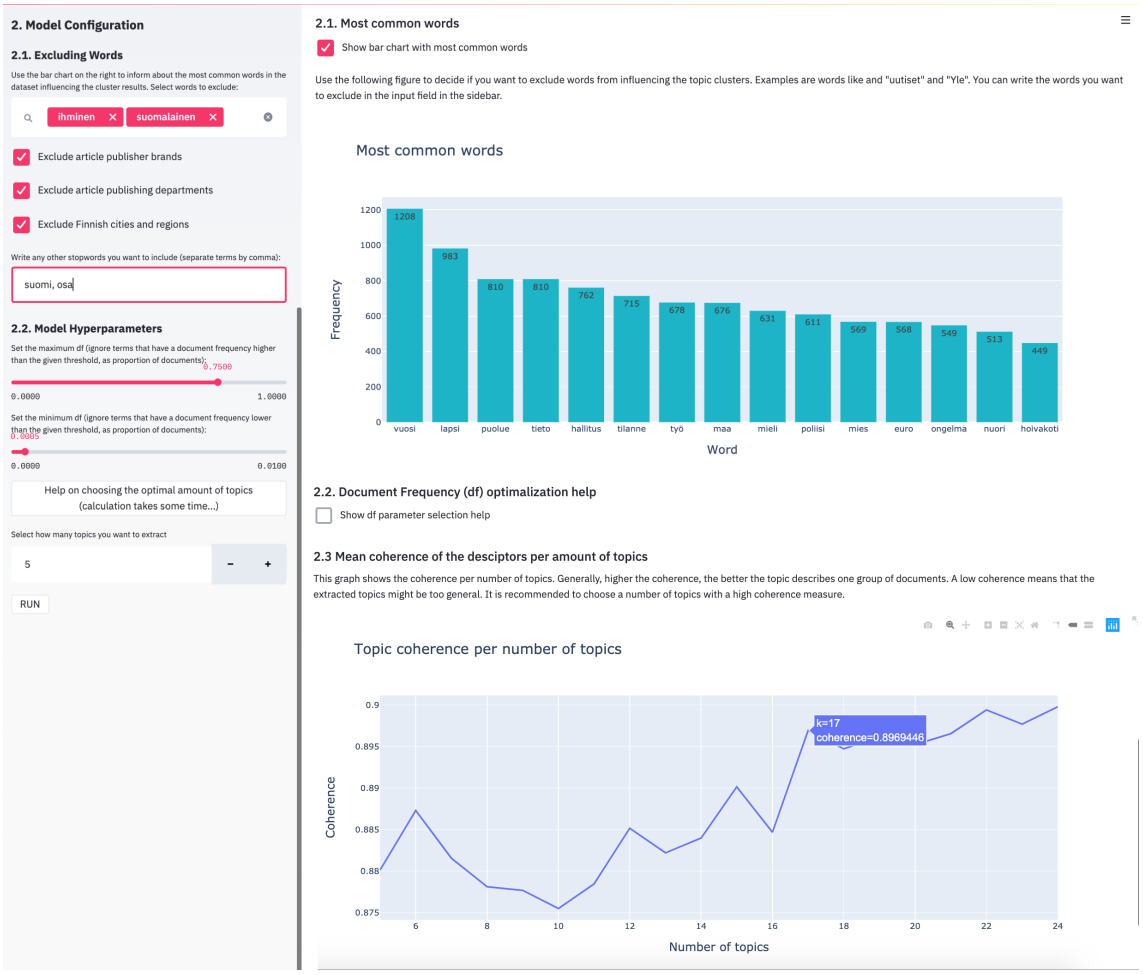

Figure 15: Screenshot of user step 2 of the ITM web application: model configuration. The left panel is for user input, in which the user can add stopwords, change tf-idf parameters, ask for topic model coherence calculations and set the number of topics to extract. The main panel visualizes the most common words and the calculated topic coherence per k.

library).

Also the relations between individual documents and their topics are visualized in a two-dimensional space, which is again only generated on user request because of the computational complexity (Figure 18). As explained in the previous section, this visualization is created using the UMAP dimensionality reduction method. The following parameters for mapping the document-topic matrix to a two-dimensional space are set:

- The number of neighbors controls how the local versus global structure in the data is balanced. The optimal number of neighbors depends on the number of datapoints, because there is a local/global trade-off. In the application, the following rule is applied (the formula is derived to have a minimum of 15 neighbors, scaling up linearly with the number of datapoints up to 200

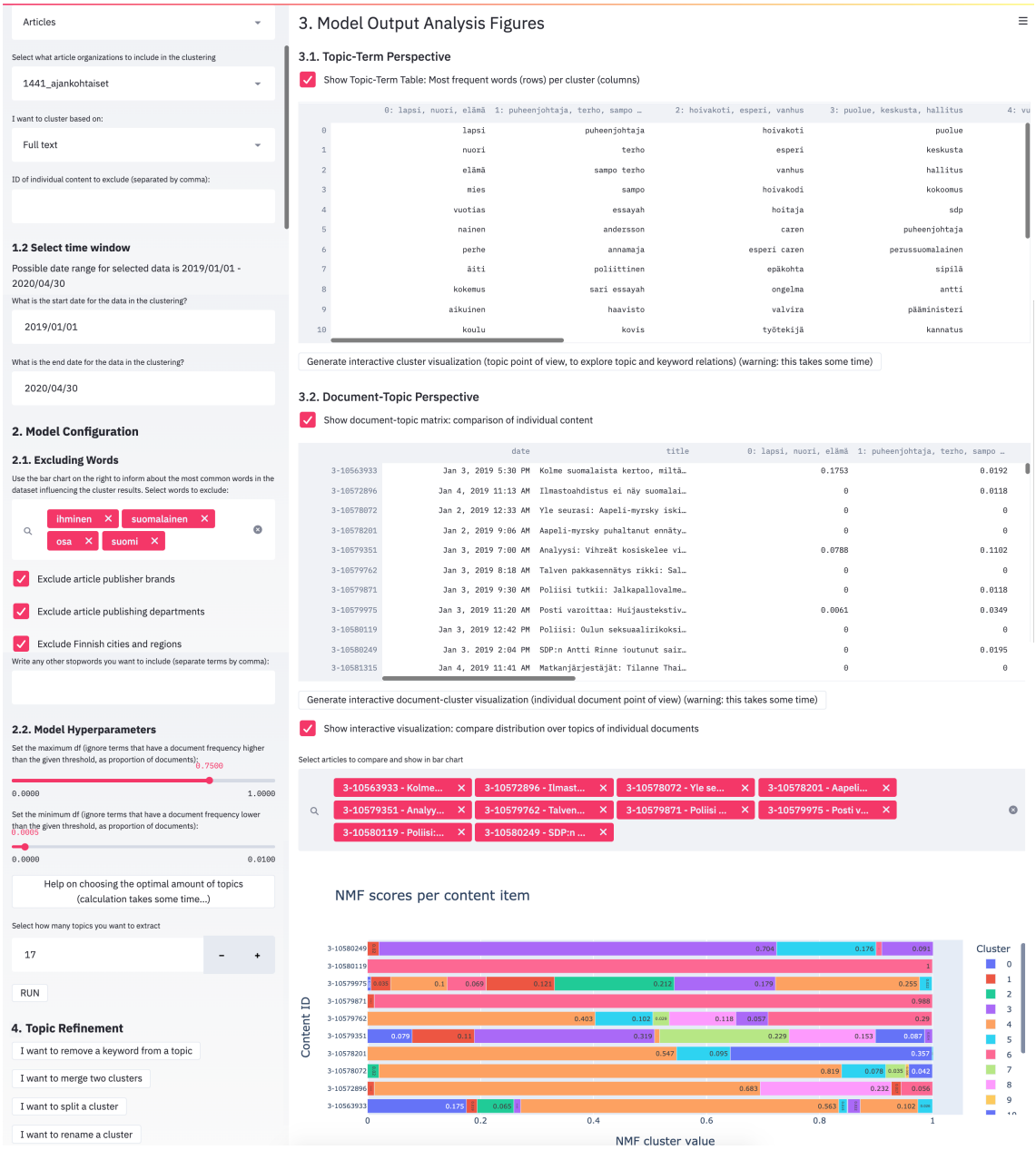

Figure 16: Screenshot of user step 3 of the ITM web application: model output figures. The model has extracted 17 topics. The output is displayed from topic-term and document-topic perspective.

neighbors with 20,000 datapoints, see Appendix D.5 for more details):

$$\text{n\_neighbors} = \max\left[(15 + 0.007 \cdot m), 200\right] \tag{4}$$

- The Hellinger distance is chosen as distance metric for computing the position of individual documents in the two-dimensional space by dimensionality reduction. This metric is custom defined (not supported by default by the UMAP Python library). The Hellinger distance quantifies the similarity between two probability

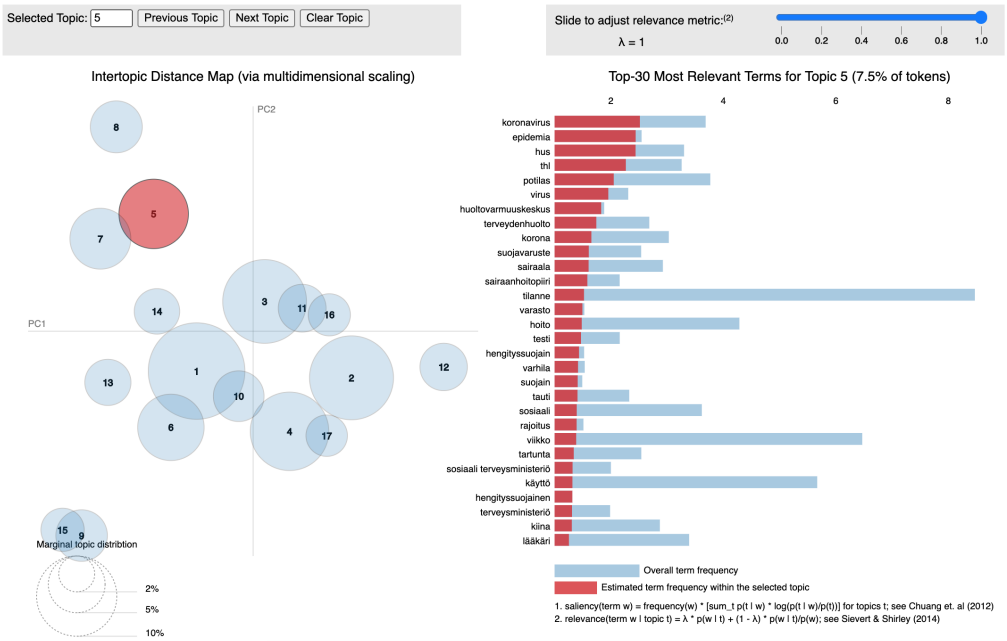

Figure 17: Screenshot of the interactive topic-term relation visualization, generated using pyLDAvis. Topic 5, which contains articles around the Covid-19 pandemic theme, is selected. The words on the right are words that are most relevant in this topic, with the light blue bars indicating the relevance of these words in the total dataset.

distributions. Since the document features are distributions of word counts generated by tf-idf, the Hellinger distance can be used to measure the similarity between these distributions. The Hellinger distance is thus more suitable in this case than (normalized) Minkowski style metrics like Euclidean or Manhattan distances or other spatial, angular or correlation metrics like the cosine distance.

Dimensionality reduction of the documents is thus performed directly on the extracted tf-idf features using the Hellinger distance metric. The computed location of the datapoints in the two-dimensional space are completely based on the feature set, and not on the output of the NMF decomposition. Only the coloring of the datapoints to the topic which is most covered in the document is a result of the topic model.

The visualization is, in contrast to all other visualizations in the application, generated using the Python Bokeh visualization library[11]. With Bokeh, a stand-alone interactive HTML figure can be generated, supporting more user interactions than visualizations created with Plotly. Bokeh is much slower in generating data plots compared to Plotly, so Plotly is preferred for all other visualizations built in the ITM application. An example of a document-topic plot generated with Bokeh is shown in Figure 18. By default, all topics and the documents belonging most to these topics are included in the two-dimensional space. The user is able to explore individual

[11] https://bokeh.org/

topics, by changing the slider, displaying only the documents that appear in this topic. When the user hovers over a datapoint, more information about the single document is presented. The user can also explore individual documents by clicking on a datapoint; the interface will highlight this document, show meta information depending on availability (e.g. title, authors, publication date), and a link to the actual item. On top of that, the user can search for documents by entering a search query in the search bar, which filters out documents that do not contain the search query.

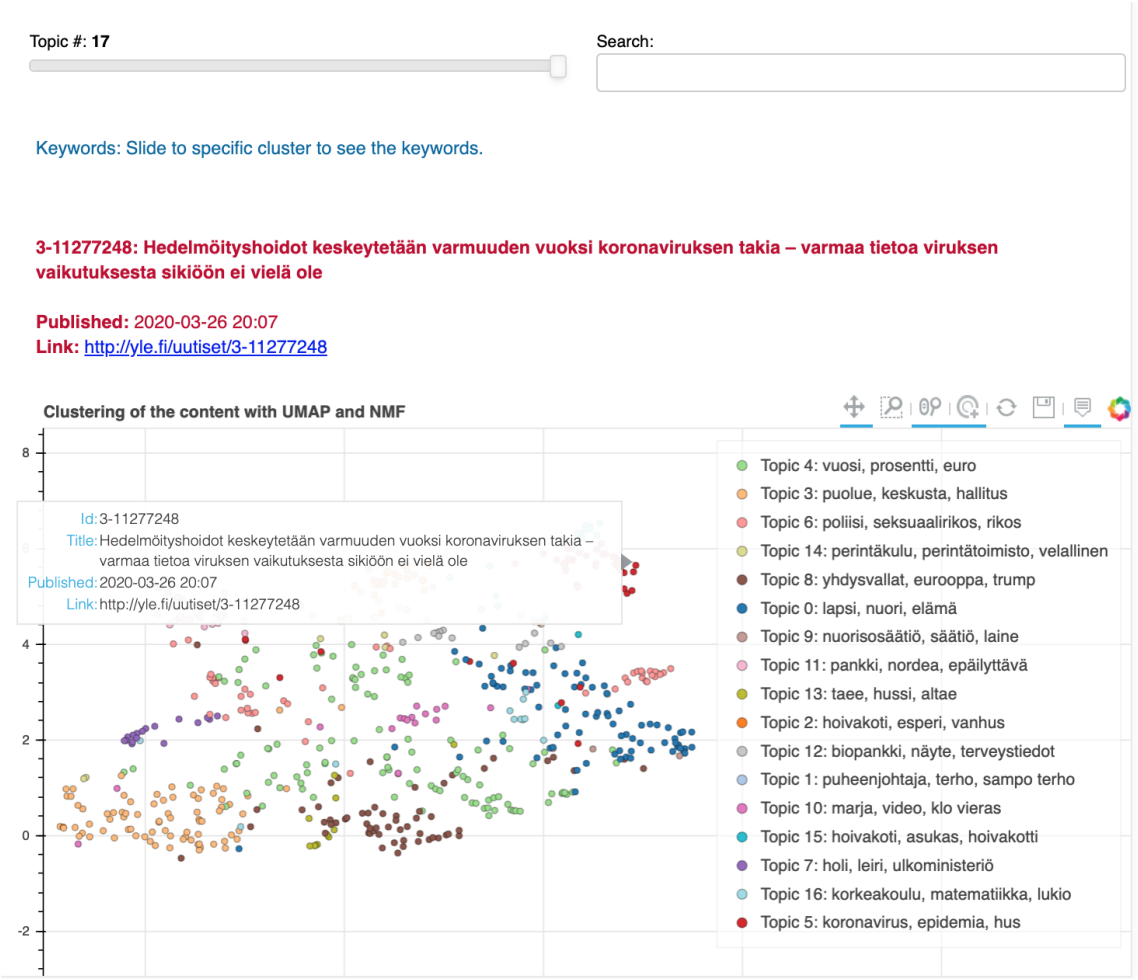

Figure 18: Screenshot of the interactive document-topic visualization, generated using Bokeh. Datapoints in the two-dimensional plots are individual articles, which have the color of the topic that is most represented in the article. One article is selected, belonging to Topic 5 about the Covid-19 pandemic (with top words coronavirus, epidemic and Helsinki University Hospital). The title, publication date and link to the actual article are presented on top. This visualization allows keyword search to individual articles and topic selection to browse through articles per topic.

Section 3.3 of the main panel (see Figure 19) displays the estimated quality per topic. The quality per topic is estimated by the taking the average residual per

topic, calculated by taking the Frobenius norm of the original document-term matrix (matrix **A**) minus the dot product of the coefficients of the calculated topics (matrix **W**) and the topics (matrix **H**) (see Appendix D.5). The higher the average residual, the worse the topic is able to represent the underlying articles, which is an indication to the user to perform some model refinement to these topics.

Figure 19: Screenshot of the estimated topic quality bar chart. The topics are sorted based on the average residual. The higher the residual, the more general the topic is, resulting in poorer performance. General performance of the topic model can be improved by refining topics with high average residual.

Section 4, model refinement options, is included in the left panel. Each possible refinement option can be triggered by clicking on a button, which opens an interactive input widget. Depending on the refinement action, topics and words can be selected from dynamically generated drop-down menus or typing. For example, selecting keywords to remove from a topic, requires the user to first select the topic to remove a word from, and then select the keyword itself. Both selections are made by drop down menus, which are dynamically generated based on the extracted topics and the topic selected by the user respectively.

Section 5 includes exploratory data analysis and the possibility to export data and settings. EDA visualization is shown in the main panel like the other model output visualizations of section 3, but is separated because these figures mainly present how the generated topics can be used in wider context, in contrast to the figures presented in the step before which are focused on topic interpretation and modeling output assessment and refinement. The figures for EDA can be accessed after the model generated the initial set of topics, but the figures are not displayed in detail. We want the user to focus on topic evaluation and refinement, before diving into how the topics can be utilized in wider context. Thus, these figures of topic development over time and comparison with audience data are only shown when checkboxes are clicked by the user (see Figure 20).

The data export functions, to export the decomposed matrices, consumption

data and user settings and model interactions, are available from the initial model results as well. Users might want to export sets of results at different moments in the modeling process. Although links to directly download content could be created in the interface, the download links will only appear after the user triggers this by a button. This extra step was implemented for performance reasons; transforming all data to CSV format on every model change takes time.

The interactive application is designed to offer a high user experience to its end-users. The non-technical domain experts are using the application to model their datasets, without in-depth knowledge about the underlying topic model. Thus, all model interaction possibilities, should be as intuitive as possible. Moreover, regardless the technical background of the end-users, all users are humans, and humans can make mistakes. Therefore, all user input is made as intuitive and error proof as possible. This implies both choices on data and model settings, as well as direct user input for model refinement. For example, warnings are shown when data input is of potential low quality and when k is too high in respect to the training data size. All data and model settings, as well as all user operations performed, can be exported so that these do not have to be remembered. In addition, model refinement actions, which require detailed user input and are not easily reversible, are only enforced in the model when the user explicitly clicks on an 'Apply changes' button. This is a way of asking confirmation to the user, makes the user aware of its actions and thus avoids unexpected actions and results. Moreover, offering drop down menus, instead of requiring typed input from the user, makes the application robust for human errors. Selecting a word from a list is less prone to human error (and thus unexpected results) than typing in a word manually.

3.3.2 Interactive Topic Model

The workflow of the topic modeling process is depicted in Figure E4 of the Appendix. The major steps of the ITM process are explained here.

Dataset creation (preprocessing outside application)

Most of the data preprocessing happens before it is used in the ITM application. Preprocessing of language data is relatively computational heavy and thus takes a lot of time. To ensure that the user does not have to wait for data to be preprocessed every time a new modeling process is initiated, and to avoid unnecessary re-preprocessing of the same data, preprocessing of the natural language is taken outside of the application workflow.

Data preprocessing consists of constructing a bag-of-words using NLP techniques. The NLP steps, as introduced in the previous section, are performed on the raw text in documents. Texts are processed with the Python library Stanza (a natural language analysis package including a pretrained neural Finnish language model,

developed by Stanford's NLP Group[12]) using the package SpaCy[13] (which wraps this library to use language models in the SpaCy NLP pipeline). The combination of these packages made all preprocessing steps possible for this application. In addition, the Finnish vocabulary library Voikko[14] was used as fallback dictionary if the pretrained neural model by Stanza was not able to parse a document text.

Additional preprocessing (inside application)

The preprocessed document corpora are directly loaded in the ITM application on request of the user. The user is then able to preprocess the data further, by removing additional (stop)words. Moreover, tuning the *document frequency* parameter in the interface influences the tf-idf feature extraction of the input data (the document-term pairs). Tf-idf feature extraction is done using the Python machine learning package Scikit-learn, with the module *sklearn.feature_extraction.TfidfVectorizer*. This module converts a collection of text documents into a matrix of tf-idf features. Default parameters[15] were used, except for the following.

- Minimum and maximum df values were determined by the user interface, with default values of 0.005 and 0.750 respectively.
- A maximum of 10,000 features will be extracted. Experience from trial rounds learned that more features takes longer to extract but did not improve the model outcome since the features are used to form only a relatively small number of topics out of many features.
- The 'sublinear tf' parameter is set to true, to apply sublinear tf scaling (i.e. tf is replaced by $1 + log(\text{tf})$). This normalizes the bias against lengthy documents versus very short documents.
- The 'ngram range' parameter is set to $(1, 3)$, so that not only unigrams would be extracted as features, but also bigrams and trigrams (terms or frequent combinations consisting of two or three single words).
- A 'token pattern' was applied, to only extract features that are words of three or more letters, containing only Unicode Regular Expressions extended by the Finnish alphabet. All other words (like numbers) are discarded.
- Manually added 'stopwords' by the user are filtered out.

NMF topic modeling

The result from feature extraction is the document-term matrix **A**. NMF decomposes this matrix into the smaller topic-term and document-topic matrices H and W. Like

[12] https://stanfordnlp.github.io/stanza/
[13] https://spacy.io/
[14] https://voikko.puimula.org/
[15] For the default parameter settings, consult
https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

feature extraction, the Scikit-learn was used for NMF decomposition in Python. The following settings were applied in the *sklearn.decomposition.NMF* module:

- The number of components to extract ('n_components') is the number of topics k, determined by the user in the interface.
- The model is initialized with Nonnegative Double Singular Value Decomposition (NNDSVD). This initialization method is better for sparseness. The matrix to decompose in this application is very sparse, since most words in the feature set are not present in most documents, and therefore a lot of zero or null values exists in the document-term matrix.
- No regularization is applied ('alpha' is set to 0).
- The 'beta loss' is set to 'Frobenius'. This measures the distance between the input matrix **A** and the dot product of the output matrices **W** and **H**, in which the beta divergence is minimized, see Formula 3.

The model is reinitialized with above settings when the user makes any changes to the data preprocessing, feature selection or number of topics to extract, and presses the button 'Run model'. When the user wants to make a refinement to the model output, like topic splitting or merging or removing a keyword from the model, a new decomposition is done, but a custom initialization is passed. The custom initialization contains a modification of the topic-term matrix from the original decomposition result, according to the user's wishes.

- **Delete keyword from topic**. The same number of topics is kept as number of components to extract. All the rows (topics) of **H** consisting of keywords with corresponding weights are also kept, except for one value. The weight of the keyword of the topic specified by the user to remove from a topic is set to zero. This new matrix **H**' is passed to the NMF model initialization. In addition, it is given that this matrix should not be updated in the new decomposition process, although the loss will most likely be higher given this custom matrix **H**'. Only the document-term matrix **W**' will be computed now, so that the user will see results as they expect. The user will thus feel in control of the model, which is important in achieving high user experience.
- **Splitting a topic**. The number of components to extract will be increased by one. A new matrix **H**' will be computed using the user's input on words to keep and discard in the new topics. The user will only be able to choose from the 20 most frequent words in a topic, so both topics will first be initialized with all the words and weights of the original single topic. Then, the weights of keywords to discard in the topics are set to zero for the specific topic. This new topic-term matrix **H**' is given as custom NMF model initialization, and only **W**' will be computed in the decomposition of **A**.
- **Merging two topics**. Merging two topics does not require new decomposition of **A**. The weights of keywords in both topics are simply added in **H** giving

H', as well as the weights of topics in documents giving **W**' out of **W**. **H**' and **W**' have one row and column less respectively, because two topics are now merged into one. No new NMF decomposition step is performed using the reference information of the merged topics in **H**' to compute **W**', because the computation would take longer and the former approach was found to empirically work better in a study by Kaung, Choo and Park [27].

Post-processing

Once the matrices **W**and **H** have been composed, some post-processing will be performed as preparation for the visualizations and better user interpretation. Topic labels are generated by taking the three most prevalent words per topic, and topics are ordered according to their size (defined by the sum of the degree of topics represented in the documents). In addition, the degree of topic representations in the documents are normalized over the topics per document (column normalization to unity of **W** by l_1-*normalization*).

3.3.3 Back-end

The main components of the back-end of the ITM application are visualized in Figure E2. As mentioned, the main application's interface and topic model are built in Python, extended with libraries for machine learning computations and visualization generation. The main Python script makes calls to Streamlit's API to build the user interface. The main Python scripts uses results from topic model computations and visualization preparations of other written scripts in Python. These modules interact with various databases containing the preprocessed document corpora, document metadata and binary Word2Vec models. Then, script runners connect the clients' application interfaces with the scripts, were changes are controlled by a file system observer. Script runners are able to save and access data in cache, which, as explained, improves performance. The runners allow multiple clients to use the application simultaneously in different browsers, using websockets and https protocols.

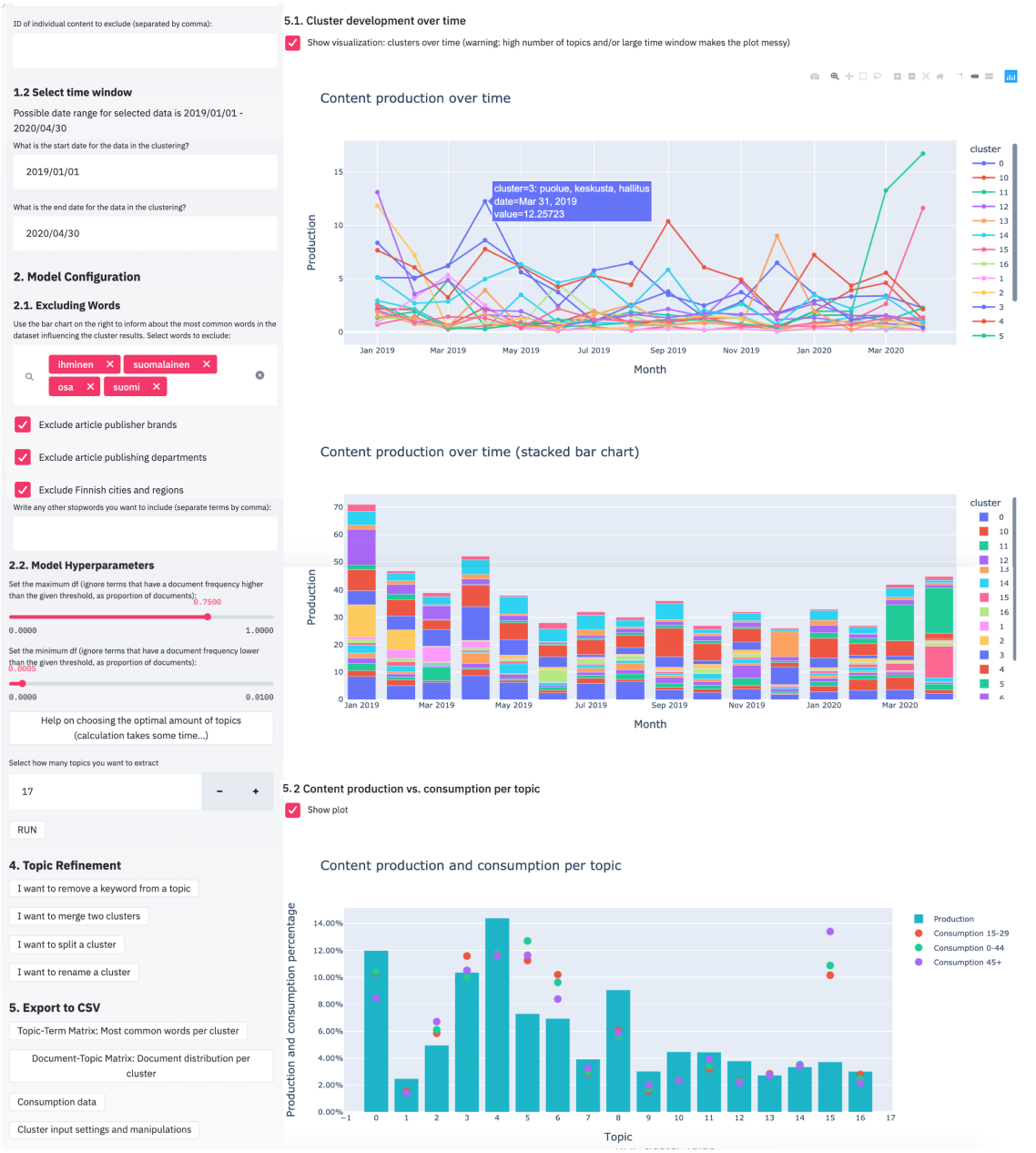

Figure 20: Screenshot of the EDA figures. 5.1 includes a line figure and a stacked bar chart for the relative content production over the selected time period. These figures clearly show the increase in production of articles about politics in spring 2019, when elections were held, and an increase in production about the Covid-19 pandemic and its cases in elderly care homes in spring 2020. Because of the high number of extracted topics, the bar chart might be more informative than the cluttered line chart. Figure 5.2 shows the relative production and consumption over time. Articles about topics 2, 5, 6 and 15 about elderly homes, coronavirus, police and coronavirus cases in elderly homes, are relatively read more than produced. This is in contrast to for example articles about economics and American politics (topics 4 and 8). In addition, topics 2 and 15, about elderly care and coronavirus in elderly homes are read mostly by the older audience (aged 45+), while topic 3 and 6 about Finnish politics and crimes are more ready by the younger audience.

4 Evaluation

The fully interactive topic modeling application is designed with and for non-technical domain experts in the multimedia company Yle. Functional testing of the application was done iteratively, at the end of every design and development cycle. Model and back-end scripts are continuously tested by unit tests, the design by integration tests, the requirements by system tests and finally the user needs by user acceptance tests (UAT). The UATs are done with participants of the focus groups during multiple sessions. Functional testing is applied to test whether the application works the way it should, but does not address the question whether the end-user can access all functions and successfully use the tool. In addition, for an application to be successful, it is not enough to be usable (i.e. users can complete defined tasks with it), but it must also be pleasant and desirable to use. A user study is conducted to evaluate if the ITM application is indeed user-friendly and efficient. In this usability test, participants are observed in their attempt to complete a topic modeling task, to reveal potential issues, uncover opportunities and assess the overall user experience. This section covers the both qualitative and quantitative evaluation of the application. First, the evaluation measures and experiment methodology will be explained. This is followed by the results from this usability and user experience study.

4.1 Methodology

4.1.1 Participants

Ten participants (6 female, 4 male), on average 30-39 years old, were recruited to take part in the evaluation study. All participants are employees at Yle, and data analysis is part of their daily job. All participants are Finnish, and speak fluent Finnish and English. The group of participants had various backgrounds regarding data analytics and data science. A short questionnaire was filled by the participants before the evaluation session to learn about their background, see Appendix G.1.

7 of the 10 participants are confident in their ability to use data analytics tools, 3 are neutral. 8 participants mention that they are confident in analyzing data, out of which 3 strongly agree to being confident. The confidence of participants' ability to use models for data prediction and machine learning varies: 3 are confident, 4 are neutral and 3 are not confident. Five participants indicated to have used a clustering model before, one has developed a clustering model before, but none of these clustering models were focused on topic modeling. Three of the participants were participants of the focus groups, and thus had an influence on the design of the application. Nevertheless, they are included in the evaluation study, because it is focused on evaluating the overall user experience, in contrast to the agile user acceptance tests, which were focused on meeting user needs and requirements.

In conclusion, a diverse group of participants took part in the evaluation study to get a broad spectrum of feedback on usability and user experience. A small sample size was used to collect subjective user experience, not aimed at generating statistically significant findings. However, subjective rating scales are complemented

with semi-structured interviews to derive useful qualitative insights into how people feel about the design.

4.1.2 Dataset and model initialization

The application includes access to various data sources, which can be filtered by metadata on top of that. For the purpose of this evaluation study, all articles published by Yle's department of Current Affairs ('Ajankohtaiset') from January 1st 2019 until April 30th 2020 were selected. This set of 605 articles, with an average of 318 words and standard deviation of 231 words in the preprocessed texts, was selected with the assumption that this dataset is familiar for the participants, thus making the topic modeling task accessible.

By default, the correct dataset and time range of data was selected. Yle specific words like brands and publishing departments (see Appendix F), as well as Finnish geographical names are excluded by default, but can be included on the users choice. The model has not run on first start up of the application, but the number of topics to derive is set to 5 (the minimum amount of topics) by default. Users can change this number before the first time they run the model. After a topic model is generated, only the topic-term and document-topic matrices are shown automatically in the main panel; all other visualizations will be shown by clicking allocated checkboxes and buttons.

4.1.3 Procedure

Each individual evaluation session took up to one hour, conducted remotely with audio and screen-capture recording. Each session started with a walk-through of the application, which introduced the participants to the application, topic modeling, supported interactions and model refinement operations, and miscellaneous application capabilities. This training was done using a dataset on video content, to simulate another use case than in the actual study.

Next, the following task scenario was presented to the participants:

"Imagine you have been asked to make a report about themes of the news articles published by the Current Affairs department ('Ajankohtaiset') since the beginning of 2019. All news articles from this department published since 2019 until the end of April 2020 have been gathered and processed, along with additional data about consumption. Use the application to derive a set of topics that would help you and your target audience in understanding what topics and themes have been covered by the articles, and how you would use this in further data or trend analysis. You can choose the number of topics, your goal is to find topics that represent the set of articles best. Use the tool to evaluate on the topics it derives, you can make refinements to the topics based on your interpretation of the topics. You do not have to make the actual data report as part of this task."

Participants were given the instructions to use the tool to complete this task, and export and send the topic model settings and data, refinements and quality when they were done. They were instructed to think-aloud during the task, to get an understanding of why they are taking certain actions and the person's reactions

and thoughts about it. Once the participants indicated that they were finished with the modeling process, a short semi-structured interview was conducted (see Appendix G.4).

After the interview, participants completed a survey containing closed questions, addressing their feeling on usability, user experience and user perception of their interaction with the application.

4.2 Measures

All user interactions with the tool are logged, along with the participants' motivation. The post-test survey contains questions to evaluate usability, user experience and user perception.

The overall perceived usability of the ITM application is measured by ten questions from the System Usability Scale (SUS) [26]. These ten questions each have five response options on a Likert scale; from 'strongly agree' to 'strongly disagree'. The questions are included in Appendix G.2. Following are four questions to measure user experience, and five questions to measure user perception. These nine subjective measures are adopted from [50], in which ITM applications were evaluated by posing participants a similar task as above. User experience measures addressed are frustration, trust, task ease and confidence. The five user perception measures are perceived adherence, perceived instability, perceived latency, final model satisfaction and perceived improvement. Respondents were able to answer again on a 5 point Likert scale from 'strongly agree' to 'strongly disagree', aside from satisfaction, which is on a scale from "not at all" to "very" and improvement, which is on a scale from "much worse" to "much better." The statements are included in Appendix G.3.

The outcomes of the think-aloud modeling sessions are coded thematically for analysis.

4.3 Results

The ten participants spent an average time of 24 minutes ($\mathbf{s}$ = 7 minutes) on the task using the ITM application.

4.3.1 SUS

The ten statements of the System Usability Scale, which measures the usability and learnability of the ITM application, were filled by the ten participants after they completed the modeling task. The average SUS score of the developed ITM application is 81, which is well above the benchmarked average for websites and web applications of 68 [32]. Figure 21 shows the distribution of the participants' answers on the ten individual statements. Note that the even items are stated reverse.

4.3.2 User Experience and User Perception

The participants' perceptions regarding adherence, instability, latency and model quality (final model satisfaction) is analyzed through subjective responses on a five-

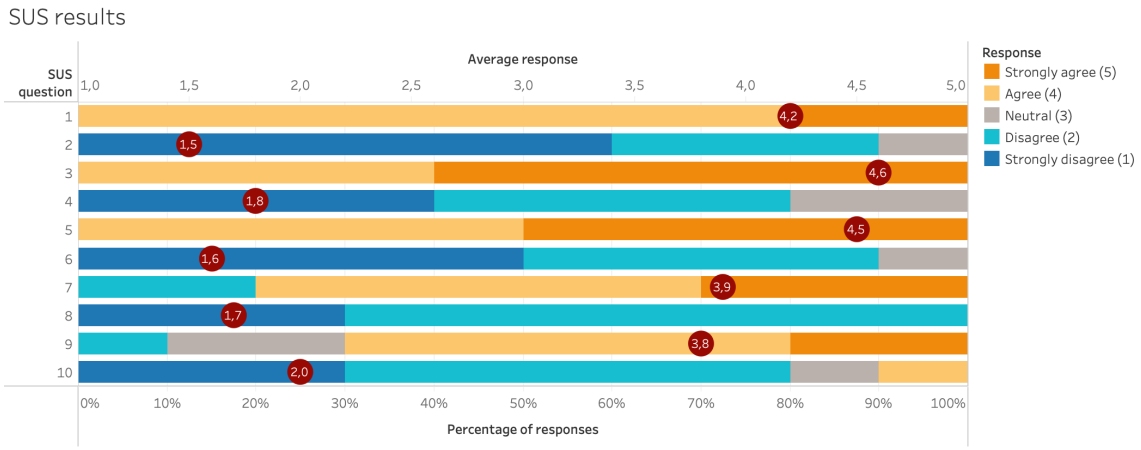

Figure 21: Average five-point rating scale responses of the SUS questionnaire per statement. Even items are stated reverse.

point Likert scale. To get an understanding on how variations in these measures affect user experience, participants also responded to four statements regarding frustration, trust, task ease and confidence. The distribution of the results are visualized in Figure 22. Participants found the task easy ($\overline{\mathbf{x}} = 4.2$, $\mathbf{s} = 0.8$), where 1 equals 'Strongly disagree' and 5 equals 'Strongly agree'), trusted the application ($\overline{\mathbf{x}} = 4.1$, $\mathbf{s} = 0.7$), were confident ($\overline{\mathbf{x}} = 3.9$, $\mathbf{s} = 0.9$) and did not experience frustration (negative statement, $\overline{\mathbf{x}} = 1.8$, $\mathbf{s} = 0.8$). The changes participants made to the topic model are overall perceived positively. They thought that the application adhered to their input ($\overline{\mathbf{x}} = 4.2$, $\mathbf{s} = 0.9$), had low latency ($\overline{\mathbf{x}} = 4.1$, $\mathbf{s} = 0.9$) and was not instable (negative statement, $\overline{\mathbf{x}} = 1.8$, $\mathbf{s} = 0.6$). In addition, 8 out of ten participants argued that the final topics were improved over the initial topics ($\overline{\mathbf{x}} = 3.8$, $\mathbf{s} = 0.4$, where 1 equals 'Much worse' and 5 equals 'Much improved'), and 9 out of 10 participants are satisfied with the model results, out of which three are very satisfied ($\overline{\mathbf{x}} = 4.2$, $\mathbf{s} = 0.4$, where 1 equals 'Very unsatisfied' and 5 equals 'Very satisfied').

4.3.3 Findings from think-aloud sessions and post-task interview

Users avoid changing complex model configuration settings before an initial run

Only two out of ten participants made changes to the default model configuration before they ran the model for the first time. Three participants made no changes at all, while five participants only increased the number of topics to extract from the initial model. P3 mentioned that she did not change any stopwords or hyperparameters because she wanted to *"see what the model comes up with before manipulating it, so that I can get an idea of how the model works and nothing important will be left out"*.

Figure 22: Five-point rating scale responses of user experience and user perception. The results show there is not much variability; participants' perceptions and experience regarding the ITM application are positive on all measured aspects.

Users are uncertain about the amount of topics to extract, but not all use the help function.

All participants mention that they are uncertain about what amount of topics to extract in the first run. Three participants decided to run the model with the default of 5, four participants changed k to 10 or 20. P5, choosing 10 topics, argues that

"just extracting 5 topics might be too few on this set of data", and P6 mentions that he *"loves to explore a lot of data first to get a general idea on what to specify in the longer run"*. Three participants use the function to estimate the coherence of different topic amounts. P2 mentions *"I don't know what amount to start with, so I hope that this estimation gives me a suggestion"*. P7 says *"I think it is difficult to decide on the optimal amount of topics before you see any results, so it really helps me that the application offers some help here"*. P7 also mentions that she *"should not extract too many topics for only one and a half year of data, but also not too few because then you might miss insights"*.

Users like to explore and try out functions multiple times

Seven out of ten participants run the model with at least two different numbers of topics to extract. P2 started with 17 topics (with the highest topic coherence measure, but tried the model with 12 (*"I think 17 topics was a bit too much, because there are two topics about politics, two topics about the coronavirus, and a really specific topic about Swedish economy"*) and 6 (*"I just want to see the difference, because the model with 12 topic looked pretty good"*) topics as well. P5 changed k 3 times, *"to see whether individual clusters will split and merge"*.

Users' behavior is very diverse

Although the scenario posed in the user study was the same for all participants, the user behavior varied in every session. Post-task interviews revealed that individual participants had different interests and use cases in mind, which resulted in differences in user actions and findings. For example, P8 focused a lot on excluding stopwords (*"I don't want common words, that can be part of any subject, to influence my modeling results."*), while P7 used multiple topic refinement operations to *"ensure topics have high quality so I can safely use them later on"*, and P6 spent most of his time exploring the topic results in the EDA figures with topic development over time and consumption rates, which *"helps in micro segment discovery"*. P4 stated: *"I have never worked with topic extraction in media, so I would evaluate the generated topics by first choosing the articles that interest me personally, and not the topics and terms themselves"*, while P6 said *"I don't think I would need to see the individual documents, I am already happy with the variety between the clusters and how they make sense to me"*.

EDA visualizations do not only serve as 'extra' analysis, but also support the users in topic model understanding, evaluation and refinement.

The exploratory data analysis visualizations is seen by P6 as *"The most interesting part, this is where I get excited"*. He mentions that *"The line graph doesn't make sense because you can't really read it, but the stacked bar chart makes a lot of sense. I think this would be the opposite when I had chosen less topics."* Four participants have used the EDA visualizations to evaluate individual topic results, and made model refinements based on analysing these graphs. P10 mentioned that she merged

two topic on the coronavirus as well as topic on politics based on trend peaks in the plot of topic development over time: *"The production of articles about Finnish politics and elections have a similar development over time, both showing a peak in spring 2019, so I will merge these two clusters together".*

Users like the variety of visualizations

Not all visualizations are used by all participants in the modeling task, but they like the diversity. For example, P3 mentioned her interest in exploring how topics are represented in individual articles, just before she discovered that this was actually possible: *"The 2D plot with links to individual articles is great, this was exactly what I was looking for, it now shows me even clearer what the topics are actually about.".*

Users like the interactivity of the visualizations

All ten participants point out that they like that the visualizations are interactive. P1 said: *"I like the interactiveness, it is really helpful to see the name of the topic when I hover over the data".* P6 agrees by saying: *"When I hover over data in the charts, I see more information on the content and the clusters, that is really good.".*

Users' curiosity and excitement is triggered and want to apply the outcome data with data from other sources

P5 said the following when exploring the last EDA graph including consumption data per topic: *"This graph really arouses my curiosity, I want to do some analysis on questions that come to mind now. It is great that I can do this analysis with just a few clicks in this application. Although it needs some more digging, I can already see that this graph proof the hypothesis that people read articles about topics that they feel identified with. For example, articles that are about the police and crimes committed by young people, are read mostly by the younger audience."* P9 mentions: *"I want to start some editorial discussions based on these last graphs. To discuss for example about what themes we wrote about last year, how they attract different kinds of audiences, and whether this follows the plan that was made last year."* P4 mentions that he learned what kind of language is used in article publications, and *"would like to compare that with articles covering the same topics coming from other sources, and combine this with ranking data of search engines".*

Users like to be in control of the model, but would like to see suggestions and quality measures

Users get excited when they see that manual refinements to the model are implemented and reflected as expected. At the same time, users also appreciate recommendations made by the application which topics to refine. This holds for topic refinement as well as model configuration settings like adding stopwords. P7 mentions: *"the residuals of individual topics make sense; the cluster about family issues is quite broad, and also the topic about Europe is vague. It is a good indication that I should split these*

topics or remove some keywords". P8 mentions: *"It is really handy to see the most common words in the dataset, this makes it intuitive and fast to decide which words to exclude"*.

Users like to see refinements reflected clearly and immediately in the output

P10, who merged and split topics, was confused where the new topics could be found, but was happy to see the results as expected when found in the topic-term matrix: *"I merged topics 2 and 15, but I wonder which topic number the result can be found in now. Ah, the topics have merged well, I am happy with the topics now, they make totally sense to me now."*. P3 indicated that *"the value of the application is that you get really quickly information on the data"*.

Users do not frequently rename topics

The option to rename a topic was only used by one out of ten participant, P2, who said: *"I want to rename cluster 11, but it does not have to be so exact because it is only for me to remember"*. P8, who did not use this option, stated *"I do not need to rename a topic, because I feel that the names already descriptive the topics well enough"*. P1 mentioned *"It is nice to have the option to change topic names, but I am not using it today because the automatically generated topic names are already sufficient for my own understanding"*.

Users without previous modeling knowledge feel confident

Participants indicated that they are amazed by how much they can achieve with this application without having any machine learning experience. They did not feel the need to completely understand the underlying model, but indicated their confidence in their actions and the final results. P2 said: *"This is so great, I am amazed, it is fascinating how much you can do with machine learning, although I don't even know how it works."*. P9, who knows about the concept of modeling but lacks experience on how to implement it, mentioned: *"I love the tool! It solves exactly the problem that I have: I want to do modeling, but I do not know how to use Jupyter notebooks or Python, so for me this is a super useful tool."*

Users like the user interface

Participants were positive about the design of the user interface. P5 said: *"The left-hand panel stays with the model settings, that is really consistent and really helpful"*. P6 said that he really liked to see that the interface designed from a user experience point of view: *"It does not only look good, but it is also very intuitive and functional, it gives a lot of details which are all very understandable"*. P9 brought up that the topic-term table could be bigger to show all topics at once. He mentions that scrolling in this table takes time and is annoying. He suggests having an overview of all topics in one view.

Users would like to see an 'undo' option.

P1 wanted to revert an action, but was not able to. *"I would like to see an 'undo' option, to revert an action if I made a mistake and want to go back to the previous model results."*

Users like to explore functionalities rather than reading the guide first, but want to see local info icons.

None of the participants conducted the 'how to use' explanation before they started modeling. *"I know what the application should be roughly capable of, so I will just explore the functionalities"* (P10). P6 proposed to introduce question marks or info buttons next to functions with explanations: *"that would help me a lot, because then I do not have to search for the information elsewhere in the application".*

5 Discussion

The knowledge gap between data scientists and domain experts in the context of broadcasting has been filled with the presented interactive topic modeling application, evidenced by SUS and user experience measures. The ITM application is designed, enabling non-technical domain experts to find and analyze latent topics in media content. Complex machine learning operations and model computations are hidden from the end-user, only showing the model's data input, output and refinement options in a simple interactive interface. The user interface is designed around five steps the user takes in the interactive topic modeling process: data input selection, model configuration, output interpretation, model refinement and EDA. All requirements gathered from focus group sessions with user groups are implemented. Positive results from the evaluation study indicate that ITM applications in other contexts or machine learning applications accessible to non-technical people in general could benefit from this design as well.

Results of the evaluative user study with ten domain experts indicate that the ITM application indeed enables topic modeling novices to perform topic modeling and manipulate the modeling process and its results, without the help of data scientists. A SUS score of 81 indicates that the applications has a high usability, falling in the top 10% of scores (90th percentile). General guidelines on the interpretation of SUS results suggest that an application with a score of 81 is perceived as having 'excellent' usability performance in the aspects of effectiveness, efficiency, overall ease of use and learnability [32]. Items 4 and 10 measure learnability of the product. Both items are stated reverse in the questionnaire. The average responses of the two items on learnability fall in the 'Strongly disagree' category (Figure 21, average of 1.9 on a five-point Likert scale), indicating that participants did not find the learnability an issue at all. This is important, since the application is intended to be used without the help of technical experts. An additional questionnaire suggests that the application achieves high user experience and users are satisfied with the model. Users felt that they improved the model based on their refinements. Although the model quality before and after user manipulations was not compared by an objective measure, the subjective measure indicates that the users perceive the application as useful. Since use cases differ per user, the model quality and final model satisfaction is best to be measured subjectively. Participants indicated that the feeling of control over the model results by the offered refinement options not only influenced the experience during the modeling, but also had a positive impact on how they perceived the final results.

The ITM application was designed taking recommendations from previous research to user behavior in ITM application in mind. Conclusions of Smith et al. [50] (Section 2.3.1) regarding users that want to be in control, dislike latency and model results should be easily interpretable, are taken in the design of the current application. Results from the evaluation study indicate that the participants indeed perceive the application pleasurable regarding these aspects. Latency was mentioned as major user experience limitation in previous applications [31]. Arapakis, Bai and Barla Cambazoglu [3] also prove that latency has effect on the user's behavior on web pages.

While the overall latency of functions in the current ITM application is low, some functions (e.g. calculating model coherence levels for different amount of topics using W2V models, or generating visualizations with dimensionality reduction) had higher computation time. Since the potential latency of these time-consuming computations is explicitly stated in the interface, participants did not mention latency as a problem. It can be concluded that not only the working of the model should be transparent to the user, but also the actions and computations embedded in the user interface should be made explicit. This avoids showing unexpected results to the user, and thus keeps the user experience high.

Previous ITM applications use various simple and more complicated figures to visualize topic model output. In this application, a combination of simple and more complex figures is implemented, to give the user the option to explore results in detail. The user study shows that participants from various backgrounds use different visualizations, depending on their personal interests. None of the participants indicated that there were too many visualizations, or that visualizations are too complicated. Some participants got triggered by the visualizations to research new use cases or refine the topic model, which indicates that leaving a variety of visualizations to the user is beneficial.

Although the ten participants of the evaluation study have various backgrounds, interests, and work domains, the results of the user experience and user perception questionnaire are stable. The average standard deviation over all questions is 0.71 on a five-point Likert scale, with the highest standard deviation of 0.9 for statements on user confidence, perceived model adherence and model latency. Thus, no big differences regarding user experience and usability are found amongst the diverse user group. It can be concluded that the ITM application is perceived as pleasant to use by a diverse, non-technical audience.

Observations of user-model interactions during the evaluation task show that participants use simple topic refinement options and avoid changing complex model settings. Complex model refinement options such as changing the keyword order and changing the weights of keywords were not implemented in the current ITM application, because they would be too complex to understand for novice users, as suggested by [31]. The user evaluation results suggest that this choice was right.

An unexpected insight is that the EDA visualizations, which were implemented as 'extra' analysis figures, support users in topic model understanding. This finding suggests that users that are able to link results to data that they are familiar with, helps in the understanding of the results and thus enables users to refine the model and its results better.

The research to the design of the application and its evaluation contributes a clearer understanding of how to design complex data modeling applications for a non-technical user group. The ITM application in this thesis is designed to extract topics from media content, for a user group of media producers, media planners and analysts, but the outcome of this research suggests that this design approach could be applied to accessible interface design of complex machine learning models. The design methodology and the five general steps of the application (Section 3.2.3) should be considered when designing a similar application in wider context.

5.1 Limitations

The research methodology and the final ITM application have some limitations.

The application developed in this research is unique in its combination of use context, data, users, interface design, topic model and visualization techniques. There are no existing interactive topic modeling applications in the broadcasting domain, and no known ITM applications that are specifically designed to be accessible to domain experts, so the performance of this application cannot be compared against other applications. In addition, the evaluation method is not adopted by other reviewed ITM applications (in other domains). This implies that the measures of the user evaluation study are indicators of usability and user experience levels of the developed application in this study, but cannot be compared to other applications. Moreover, the use case company Yle did not use topic modeling before, so there is no baseline to compare to.

Second, the application is designed to extract and visualize a relatively small number of topics from a set of documents. Visualizations in the applications are designed to represent results of up to 15 topics, and can become hard to interpret or slow to generate when used on a larger number of extracted topics. Although the underlying NMF topic model is not affected (still achieving low latency and high performance), the representation of results in the user interface is. For example, the line diagram of topic development over time becomes rather messy when a large number of topics is chosen. Participants of the first focus group sessions indicated their interest in extracting up to 15 topics. Focus group sessions later in the design process revealed use cases which would benefit from extracting a larger number of topics (up to 50).

5.2 Recommendations for future work

The evaluation study outcomes and limitations lead to recommendations for future research.

First, there are a few design recommendations for future ITM applications. They are derived from the user evaluation study in combination with recommendations from previous studies. First, 'undo' should be supported, so that users are able to revert to prior states of the model. This is suggested by participant 1 in the user study, and is in line with a user study by Smith et al. [52]. Second, it is recommended to implement information about complex functions and model interactions by local info buttons at the specific function location. The user study shows that users tend to skip the general 'How to use' guide containing all information and tips, and start using the model straight away. It could however be beneficial if they could quickly access information and hints per functionality, so they do not have to search for it elsewhere. This is suggested by P10 in the user evaluation study. Third, it is recommended to allow multi-word refinement. Currently, users can only remove words from a topic one by one, but P4 indicated that multi-word removal would improve the usability. A fourth potential improvement, suggested by P9 of the user study to the current application design concerns how tables represent results.

Currently, not all topics can be viewed without scrolling in the topic-term table without scrolling (if the number of extracted topics is larger than 5). In addition, the document-topic table represents the document numbers and titles, but the titles are not fully visible and are only present in the first column, which makes it hard to compare with topic distribution numbers of high numbered topics represented in columns at the right side of the table. Recommended is to implement links to the documents, as well as an option to sort how columns (topics) are sorted from left to right, to improve the usability and user experience of interacting with this table. Finally, the visualizations presented in the interface are designed for topic modeling on a small number of topics. Tables and figures should be optimized to visualize a large number of extracted topics, if desired for future use cases.

Currently, there is only one version of the application with all its components. But when there are different methods for the same functionality, these could be tested using for example A/B studies. An example is to improve the automatic labeling of the topics. The labels could potentially be improved by taking the semantic center of the top keywords, instead of listing the three most frequent words per topic. In addition, keywords in topics are currently ranked according to their probability of occurrence in the topics. This method is not evaluated against two other keyword ranking methods: by relevance and saliency. Research could be done to find out which method is most intuitive to users. In addition, the application could

Moreover, the current user study evaluates how users perceive and use the ITM application itself. The development of this application falls in the wider goal of using extracted topics in combination with other media data sources for exploratory data analysis. Topics and distributions over documents can be exported from the application, to enable data analysis outside of the ITM interface. How users would use these topics in other data analysis applications like Microsoft Excel or Tableau is not researched or evaluated. Recommended is a future user study to how people would use extracted topics outside the application, to find out if the extracted topics are actually useful.

In addition, the application is designed to use in the multimedia content production domain, and evaluated with participants with expertise in this domain. It is not clear if user groups in other industries, such as healthcare, bioinformatics, and advertisement, could benefit from such an application. A similar user study could be conducted with participants from different domains to investigate its potential.

Then, the interactive topic model method for topic splitting could be improved. Topic splitting currently only changes the weight of the top 20 keywords, divided into two topics by the user. This could be improved by for example setting all other weights to zero. Another, potentially better, solution is to automatically determine which other words are the semantically closest in each new topic. This however requires consulting the W2V model (or any other model considering word semantics), which increases the computation time.

The application could be extended by providing topic distribution predictions of new documents. This is not featured as interactive function in the current application because this falls outside the framed interactive topic modeling scope, but could be a valuable feature for future usage. Using the fitted NMF model, topic distributions

could easily be predicted from unseen documents.

Additionally, it is advised to do future research to improve the underlying topic model. The current topic model does not take care of lexico-semantic, syntactic, or orthographic variations in text. In addition, the current feature selection method (tf-idf) ignores the order of words in a document and suffers from high dimensionality. Recent research introduces alternative feature selection methods that capture word order and context, which could improve performance of feature selection for topic modeling. For example taxonomy-augmented features leverage semantic word embeddings by taking the word order into account and reducing the dimensionality [47]. Another possibility for improvement of the topic model is filtering out named entities from raw texts in the preprocessing phase. Applying Named Entity Recognition (NER) in preprocessing enables to filter out locations and names of people, places and instances automatically. In the scope of this thesis, this step was performed manually. It is questionable whether filtering out all named entities from all documents is desired in topic modeling of media; focus group participants indicated that they want to have input on what named entities to keep.

Finally, the topic modeling method NMF was implemented based on literature research findings, and is not evaluated separately in this thesis. Literature [15] shows that a hybrid method of NMF and pLSA leads to significant improvements over NMF-only or pLSA-only methods. A hybrid version could be investigated to improve topic modeling results in the current application.

6 Conclusion

This research has filled the gap between domain experts lacking data science knowledge and data scientists lacking domain knowledge, in the field of topic modeling of media content. An interactive topic modeling application for multimedia content production analysis has been designed, developed and evaluated. The ITM application enables non-technical domain experts of the Finnish broadcasting company Yle to perform topic modeling on their published media content, without the help of data scientists. The application consists of an interactive user interface, hiding complex machine learning model components like the NMF topic model, NLP preprocessing methods and Word2Vec models, in the background.

The application was designed with focus groups consisting of end-users (domain experts), using the human-centered design method. Requirement analysis with these focus groups in combination with recommendations from previous (I)TM applications led to a series of design choices. The underlying topic model is based on non-negative matrix factorization, which has low latency, high interpretability, high consistency and high output quality. The user interface consists of five main components, designed to guide the user through the modeling process as intuitively as possible. The user starts with selecting a data source and configuring the model by manually adding stopwords and choosing the number of topics to extract. Once the model has run, the user can use various figures from topic, document and term perspective to interpret and evaluate the results. Refinements to the extracted topics can be made: keywords can be removed, topics can be merged or split, and topics can be renamed. Finally, interactive exploratory data analysis figures utilizing extracted topics are presented; users can view topic development over time, and compare the production and consumption numbers of documents in topics. Additionally, users are supported in the choice of the model interactions by estimates on model quality, which indicates the optimal number of topics to extract from the chosen dataset, as well as the quality of individual topics suggesting the user which topics to refine.

Based on the evaluative user study, it can be concluded that this ITM application design has high usability, high user experience and is easy to learn by non-technical domain experts. Users reported the application as very user-friendly, effective and efficient.

This research contributes to the field of interactive topic modeling, but also to machine learning accessibility in bigger perspective. The gap between domain experts and data scientist is clear in broadcasting companies with media content produced and planned by journalists, but can also be observed in other domains. The design approach for the ITM application developed in this thesis (the five user interface components as result of the HCD process) can serve as interface design guideline for machine learning models used by non-technical domain experts. This thesis shows that machine learning models can be made more accessible by hiding low-level details of complex models, and translating them into high-level model interactions in a user interface instead.

References

[1] Behnoush Abdollahi and Olfa Nasraoui. "Transparency in fair machine learning: The case of explainable recommender systems". In: *Human and Machine Learning*. Springer, 2018, pp. 21–35.

[2] Eric Alexander and Michael Gleicher. "Assessing topic representations for gist-forming". In: *Proceedings of the International Working Conference on Advanced Visual Interfaces*. 2016, pp. 100–107.

[3] Ioannis Arapakis, Xiao Bai, and B Barla Cambazoglu. "Impact of response latency on user behavior in web search". In: *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 2014, pp. 103–112.

[4] Sanjeev Arora, Rong Ge, and Ankur Moitra. "Learning topic models–going beyond SVD". In: *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*. IEEE. 2012, pp. 1–10.

[5] Aneesha Bakharia et al. "Interactive topic modeling for aiding qualitative content analysis". In: *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval*. 2016, pp. 213–222.

[6] Kathy Baxter, Catherine Courage, and Kelly Caine. *Understanding your users: a practical guide to user research methods*. Morgan Kaufmann, 2015.

[7] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.

[8] Gerlof Bouma. "Normalized (pointwise) mutual information in collocation extraction". In: *Proceedings of GSCL* (2009), pp. 31–40.

[9] Allison June-Barlow Chaney and David M Blei. "Visualizing topic models". In: *Sixth international AAAI conference on weblogs and social media*. 2012.

[10] Jonathan Chang et al. "Reading tea leaves: How humans interpret topic models". In: *Advances in neural information processing systems*. 2009, pp. 288–296.

[11] Jaegul Choo et al. "Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization". In: *IEEE transactions on visualization and computer graphics* 19.12 (2013), pp. 1992–2001.

[12] Jason Chuang, Christopher D Manning, and Jeffrey Heer. "Termite: Visualization techniques for assessing textual topic models". In: *Proceedings of the international working conference on advanced visual interfaces*. 2012, pp. 74–77.

[13] Dai Clegg and Richard Barker. *Case method fast-track: a RAD approach*. Addison-Wesley Longman Publishing Co., Inc., 1994.

[14] Scott Deerwester et al. "Indexing by latent semantic analysis". In: *Journal of the American society for information science* 41.6 (1990), pp. 391–407.

[15] Chris Ding, Tao Li, and Wei Peng. "On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing". In: *Computational Statistics & Data Analysis* 52.8 (2008), pp. 3913–3927.

[16] Susan T Dumais et al. "Using latent semantic analysis to improve access to textual information". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1988, pp. 281–285.

[17] Li Fei-Fei and Pietro Perona. "A bayesian hierarchical model for learning natural scene categories". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 2. IEEE. 2005, pp. 524–531.

[18] Matthew J Gardner et al. "The topic browser: An interactive tool for browsing topic models". In: *Nips workshop on challenges of data visualization*. Vol. 2. Whistler Canada. 2010.

[19] Derek Greene, Derek O'Callaghan, and Pádraig Cunningham. "How many topics? stability analysis for topic models". In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer. 2014, pp. 498–513.

[20] Thomas L Griffiths and Mark Steyvers. "A probabilistic approach to semantic representation". In: *Proceedings of the annual meeting of the cognitive science society*. Vol. 24. 24. 2002.

[21] Thomas Hofmann. "Probabilistic Latent Semantic Analysis". In: *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*. UAI'99. Stockholm, Sweden: Morgan Kaufmann Publishers Inc., 1999, pp. 289–296. ISBN: 1558606149.

[22] Enamul Hoque and Giuseppe Carenini. "Convisit: Interactive topic modeling for exploring asynchronous online conversations". In: *Proceedings of the 20th International Conference on Intelligent User Interfaces*. 2015, pp. 169–180.

[23] Yuening Hu et al. "Interactive topic modeling". In: *Machine learning* 95.3 (2014), pp. 423–469.

[24] *Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems (ISO 9241-210:2019)*. Standard. London, UK: British Standards Institution, July 2019.

[25] Michael I Jordan et al. "An introduction to variational methods for graphical models". In: *Machine learning* 37.2 (1999), pp. 183–233.

[26] Patrick W Jordan et al. *Usability evaluation in industry*. CRC Press, 1996.

[27] Da Kuang, Jaegul Choo, and Haesun Park. "Nonnegative matrix factorization for interactive topic modeling and document clustering". In: *Partitional Clustering Algorithms*. Springer, 2015, pp. 215–243.

[28] Todd Kulesza et al. "Explanatory debugging: Supporting end-user debugging of machine-learned programs". In: *2010 IEEE Symposium on Visual Languages and Human-Centric Computing*. IEEE. 2010, pp. 41–48.

[29] Jey Han Lau, David Newman, and Timothy Baldwin. "Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality". In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. 2014, pp. 530–539.

[30] Hanseung Lee et al. "iVisClustering: An interactive visual document clustering via topic modeling". In: *Computer graphics forum*. Vol. 31. 3pt3. Wiley Online Library. 2012, pp. 1155–1164.

[31] Tak Yeon Lee et al. "The human touch: How non-expert users perceive, interpret, and fix topic models". In: *International Journal of Human-Computer Studies* 105 (2017), pp. 28–42.

[32] James R Lewis and Jeff Sauro. "Item benchmarks for the system usability scale". In: *Journal of Usability Studies* 13.3 (2018), pp. 158–167.

[33] Lin Liu et al. "An overview of topic modeling and its current applications in bioinformatics". In: *SpringerPlus* 5.1 (2016), p. 1608.

[34] Laurens van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE". In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[35] Leland McInnes, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction". In: *arXiv preprint arXiv:1802.03426* (2018).

[36] Jaimie Murdock and Colin Allen. "Visualization techniques for topic model checking". In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015.

[37] David Newman et al. "Evaluating topic models for digital libraries". In: *Proceedings of the 10th annual joint conference on Digital libraries*. 2010, pp. 215–224.

[38] Jakob Nielsen. "Enhancing the explanatory power of usability heuristics". In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 1994, pp. 152–158.

[39] Kamal Nigam et al. "Text classification from labeled and unlabeled documents using EM". In: *Machine learning* 39.2-3 (2000), pp. 103–134.

[40] Richard Nurse. *Data and analytics skills assessment*. Mar. 2016. URL: http://www.open.ac.uk/blogs/LibraryData/?p=151.

[41] Derek O'callaghan et al. "An analysis of the coherence of descriptors in topic modeling". In: *Expert Systems with Applications* 42.13 (2015), pp. 5645–5657.

[42] Pentti Paatero and Unto Tapper. "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values". In: *Environmetrics* 5.2 (1994), pp. 111–126.

[43] Jonathan K Pritchard, Matthew Stephens, and Peter Donnelly. "Inference of population structure using multilocus genotype data". In: *Genetics* 155.2 (2000), pp. 945–959.

[44] Christian Rohrer. “When to use which user-experience research methods”. In: *Nielsen Norman Group* (2014).

[45] Amir M Saeidi et al. “ITMViz: interactive topic modeling for source code analysis”. In: *2015 IEEE 23rd International Conference on Program Comprehension*. IEEE. 2015, pp. 295–298.

[46] Michael Sedlmair, Miriah Meyer, and Tamara Munzner. “Design study methodology: Reflections from the trenches and the stacks”. In: *IEEE transactions on visualization and computer graphics* 18.12 (2012), pp. 2431–2440.

[47] Sattar Seifollahi et al. “Taxonomy-Augmented Features for Document Clustering”. In: *Australasian Conference on Data Mining*. Springer. 2018, pp. 241–252.

[48] Carson Sievert and Kenneth Shirley. “LDAvis: A method for visualizing and interpreting topics”. In: *Proceedings of the workshop on interactive language learning, visualization, and interfaces*. 2014, pp. 63–70.

[49] Svenja Simon et al. “Bridging the gap of domain and visualization experts with a liaison”. In: *Accepted at the Eurographics Conference on Visualization (EuroVis 2015, Short Paper)*. Vol. 2015. The Eurographics Association. 2015.

[50] Alison Smith-Renner et al. “Digging into user control: perceptions of adherence and instability in transparent models”. In: *Proceedings of the 25th International Conference on Intelligent User Interfaces*. 2020, pp. 519–530.

[51] Alison Smith, Sana Malik, and Ben Shneiderman. “Visual analysis of topical evolution in unstructured text: Design and evaluation of topicflow”. In: *Applications of Social Media and Social Network Analysis*. Springer, 2015, pp. 159–175.

[52] Alison Smith et al. “Closing the loop: User-centered design and evaluation of a human-in-the-loop topic modeling system”. In: *23rd International Conference on Intelligent User Interfaces*. 2018, pp. 293–304.

[53] Alison Smith et al. “Concurrent visualization of relationships between words and topics in topic models”. In: *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*. 2014, pp. 79–82.

[54] Alison Smith et al. “Evaluating visual representations for topic understanding and their effects on manually generated topic labels”. In: *Transactions of the Association for Computational Linguistics* 5 (2017), pp. 1–16.

[55] Keith Stevens et al. “Exploring topic coherence over many models and many topics”. In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics. 2012, pp. 952–961.

[56] Vassilios Tzerpos and Richard C Holt. “MoJo: A distance metric for software clusterings”. In: *Sixth Working Conference on Reverse Engineering (Cat. No. PR00303)*. IEEE. 1999, pp. 187–193.

[57] Jarke J Van Wijk. "Bridging the gaps". In: *IEEE Computer Graphics and Applications* 26.6 (2006), pp. 6–9.

[58] Jun Wang et al. "Interactive Topic Model with Enhanced Interpretability." In: *IUI Workshops*. 2019.

[59] Xing Wei and W Bruce Croft. "LDA-based document models for ad-hoc retrieval". In: *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 2006, pp. 178–185.

[60] B Xie, L Song, and H Park. "Topic modeling via nonnegative matrix factorization on probability simplex". In: *NIPS workshop on topic models: computation, application, and evaluation*. 2013.

[61] Chunyao Zou and Daqing Hou. "LDA analyzer: A tool for exploring topic models". In: *2014 IEEE International Conference on Software Maintenance and Evolution*. IEEE. 2014, pp. 593–596.

A Stakeholders

From desk research, the following stakeholders were identified:

1. **End-users** of the ITM application: domain experts within the broadcasting company. They are data analysts: affinity for data and describe themselves as data-literate, but do not have an engineering background. Part of their daily job is analysing data and making statistical reports, with the help of spreadsheet and interactive data visualization tools like Microsoft Excel and Tableau Software. The national broadcasting company Yle is the work domain of all end-users, which means data they interact with is mostly about content (article, audio and video) production and consumption across different types of media platforms (television, web pages, mobile applications, social media, etc), by various types of audience (with different demographics, media consumption intentions and behavior, etc).

2. **Content producers and journalists**: data-literate domain experts who are interested in learning on what topics represent their produced media. They do exploratory data analysis on various sources of data and will consume the output of the content topic modeling, but will not directly interact with the interactive topic modeling application itself.

3. **Production decision-makers and media planners**: who mostly want to be informed about new insights and trends that may influence their decision making process. They will not interact with the application itself, but will consume the results of the subsequent data analysis.

4. **Managers** of the direct users: who play a role from a business point of view. They make decisions based on data reports, and thus benefit from the results of the ITM application, given that new insights can be gained. In addition, managers may decide on where their teams spend time on, and thus whether an ITM application may be used in the first place. It is important to assure buy-in from this group of stakeholders.

5. **System administrators and IT**: who offer the infrastructure for deploying and maintaining the product.

6. **Data scientists and engineers**: who are responsible for the technical model, implementation and improvements of the application. Technical questions from all stakeholders about the product will be taken by this group of people.

7. **Product owner**: who is responsible for the final product, its maintenance and marketing.

The first group, the direct end-users of the application, is the only primary stakeholder group. The data-literate producers and journalists mentioned in the second group are secondary users, since they either never or only infrequently interact with the product itself. This group rather uses the product through an intermediary,

which is the first mentioned user group. Then, the other stakeholder groups are tertiary users. These users are affected by the system and/or decision-makers. Note that, in this context, individual people can belong to multiple stakeholder groups, since people may have multiple roles in the company. For example, a content producer (group 2) may also be a production decision maker or media planner (group 3). Designing the application is done with taking the different *stakeholder* groups in mind, rather than different individual persons, since their roles, and thus desires, can be ambiguous.

B Focus group sessions

B.1 First focus group session

The first focus group consisted of two potential end-users, and one secondary user. One end-user is an analyst at the Yle News Lab[16], while the second primary user is a data analyst and head of Yle's general data analytics team. The secondary user is web producer at the Yle department of current affairs. None of the participants have worked with any topic modeling application tool before. The session took one hour, and was held in-person.

An open discussion about possibilities and desires was facilitated. A flexible protocol served as logical questioning route, to trigger natural exchange of ideas and information. After an initial, open discussion on opportunities and wishes from the participants, various model, interaction, visualization and design possibilities were proposed. This informed the participants what is possible from a technical and design point of view. Participants indicated their preferences amongst the options, which led to a richer set of requirements.

User requirements are summarized in Appendix C. This list includes all requirements that were gathered, thus not only those extracted from the first focus group session, but also those from iterations and theoretical background. All requirements are labeled with one of three categories: application architecture (*1, business*), technical topic model implementation (*2, system*), and user interface and interactions (*3, design*). Finally, the requirements are prioritized using the MoSCoW method, into four types: *Must have* (crucial), *Should have* (important and recommended but not crucial), *Could have* (nice-to-have, implement if time allows) or *Will not have* (Not recommended; requirements that would improve the system but are outside the scope of the current study) [13].

In addition to the requirements listed in Appendix C, additional attitudes and desires were observed during the first focus group session that should be considered in the application design:

- The end-users would like to perform the topic modeling 'every few weeks to months', but multiple times with various subsets of a bigger dataset of articles. They want to filter on metadata to achieve this (for example the type of the articles, publishing department of the articles, etc).

- The end-users are able to spend time on the total modeling process (from opening to closing the application after completion). As long as the end-user is able to extract meaningful and accurate topics, it is worth the time to explore and refine results.

- The preferred number of topics or themes to extract lies between 5 and 15. This was later updated to 5 to 50 topics, after conducting other stakeholders.

[16] Yle News Lab (`https://newslab.yle.fi/`) is a testing laboratory for journalism. Through an agile way of working, ideas to renew the way news and journalism are conducted and the services they are consumed in are developed and implemented. For more information, see newslab.yle.fi.

- The end-users would like to see some visualization which helps to quickly determine what stop words to exclude. It is mentioned that it is hard to come up with words to exclude from modeling without seeing which words influence the modeling process.

- The end-users, as well as the secondary users, would like to see preliminary visualizations on how the extracted topics can be used. This helps the primary as well as secondary users determine if the extracted topics are meaningful. Proposed was a visualization how topics are represented in the set of input documents over time, participants would like to see this included in the application.

- The end-users are willing to spend time and effort to interact with the model for the best results, but point out that they would rather *refine* settings that are proposed by the model than manually set parameters from the beginning themselves. This applies to model parameters as well as topic labeling. Derived from this is that the application should propose settings and output labels, and give the user the opportunity to edit these proposed settings and labels.

- The design of a product is not limited to the requirements of the end-users. It became clear that the results of the topic modeling are going to be used by the secondary users: the 'data-literate producers'. Although this user group is not interacting with the modeling application itself, the analysts are using the results of the model, the extracted topics, in exploratory data analyses (EDA). Since the primary goal of this group is to gather new insights from EDA of the media topics combined with other data sources, it is important that extracted topics are accurate, meaningful and easy to understand. Additionally, the learned topic should be extracted to a data format that is easy to handle by other data analysis software.

B.2 Recurring focus group sessions

Recurring sessions with two focus groups followed the first focus group session and the first design iteration:

1. The same focus group as the initial user requirement research, with two end-users and one secondary user. Sessions with this group were held every one or two weeks for the course of the two-month design and development cycle. This group had a very active role in the participatory design process. Also concept testing and desirability study methods were applied in some sessions.

2. A focus group with three end-users, one secondary users and two tertiary users. Sessions with this focus group were less frequent, and started later in the design process. Sessions were less oriented to design of the platform, but more towards concept and usability testing. Although only a small amount of new requirements were extracted from this group of participants, they confirmed

findings from the other focus group sessions. In addition, addressing a second focus group leads to a design for a wider target group and avoids design fixation.

No strict protocol was followed during these sessions. Study methods that were applied during these sessions varied, depending on the design status (the phase in the HCD cycle). In the beginning of the design and development process, sessions are more focused towards participatory design, whereas later more desirability studies tend to be conducted.

An example of a **participatory design** session is finding out whether and which model revision techniques are desired. A prototype of the ITM application was presented to the participants. Participants were asked to interact with a high-fidelity prototype of the first part of the ITM application, in which no revision interaction was possible. The users were thus presented with a topic modeling output, and were asked whether they would like to make any changes to the result, and which changes. After an initial discussion on the users' wishes, the range of possibilities were presented, to give the participants the opportunity to think about those as well.

An example of applying the **concept-testing method** in a focus group session is introducing possible topic model visualizations to the end-user. In early focus group sessions, users are presented with different types of (graphical and textual) visualizations that represent (part of) the model output. These visualizations are gathered from topic models introduced in Section 2.2, supplemented with other possible (interactive) data visualizations. Participants are asked to give their preference on visualizations, which they would most use and benefit from during the interactive modeling process. This method is used to understand if the users would want or need specific visualizations.

Finally, **desirability study** techniques are applied in some focus group sessions as well. Although desirability studies are principally carried out to find out what visual design alternative is preferred, the method is applied here to find out the most desired option in a set of presented designs in general. An example where this is applied is on interactive versus non-interactive visualizations. Data plots that represent the same data but differ in whether they provide interactivity (for example selecting a subset of data to display), are both presented to users, after which they are asked to indicate their preference.

The results of these sessions are incorporated in Appendix C.

Additionally, the following was observed:

- From exploratory data analysis with topic modeling results from testing sessions, the stakeholders showed most interest in two data analysis visualizations. First, how different topics in produced media content relates to its consumption over different age groups, and second, how the representation of topics in produced media develops over time.

- Although wishes, desires and requirements from both focus groups were mostly complementary, sometimes they conflicted. An example is the number of topics to extract. Group 1 indicated that extracting up to 15 topics would suffice, while group 2 indicated interest in extracting a larger number of topics

(towards 50). The possible number of topics influences how the model should be configured for optimal results, constraints on the data input (the larger the number of topics, the larger the number of input documents need to be to extract meaningful topics), and, lastly, influences output visualizations. For example, increasing the number of extracted topics to 50 makes visualizations that plot topic development over time messy, while up to 15 topics are clear to distinguish in a single plot. Conflicting requirements like this were solved by discussing the concerning requirement with both focus groups, and a decision was made keeping the technical requirements and capabilities in mind.

- The iterative nature of the design and development process also led to revisions of requirements within the groups. While this is the advantage of HCD, this may also lead to conflicting desires, and requires a thought-out decision. This may slow down the development, but believed is that this will only benefit the end product.

#	Cat.	Prio	Requirement description	Justification
RM1	1	M	The platform should be accessible at any time through the browser	No need to download software
RM2	1	M	The platform should be reliable and errorless	The application should not fail during the modeling process, because this can be annoying and confusing
RM3	2	M	The articles/videos should have a distribution over clusters as output (output topic-document distribution, no hard clustering)	Multiple topics can be represented in one document
RM4	2	M	Option to split one topic	Sometimes a topic contains two topics from user point of view, so ability to split
RM5	2	M	Topics do not have to be dynamically created over time	Dynamic topic modeling over time results in multiple sets of topic distributions. Users want only one set of topics over the chosen time window.
RM6	3	M	Option to choose the data source	There are multiple datasources in Yle
RM7	3	M	Option to filter the data source by metadata	Modeling on subsets of data by e.g. publication department, genre
RM8	3	M	Option to choose the time window of the data source	
RM9	2, 3	M	Option to extract (1) 5-15 topics, (2, updated) 5-50 topics	Number of topics to extact differs per use case and size of dataset
RM10	2	M	Possibility to embed new documents with the extracted topics	
RS1	2	S	Option to extract the topics with the identifying top occurring words (topic-term extraction)	
RS2	2	S	Option to extract the input document set with their topic distributions (document-topic extraction)	
RS3	2	S	Option to merge two topics	
RS4	3	S	The platform should show results quickly	"I don't want to wait too long to see the results, and I want to see them directly"
RS5	3	S	The data source should be up-to-date (should contain content that was produced until the day before)	
RS6	3	S	Option to exclude individual documents	
RS7	3	S	The application should be self-explanatory	Any domain expert and data analyst should be able to understand how to use it
RS8	3	S	Option to save custom modeling settings	So that the user does not have to remember
RS9	3	S	Option to save refinement history	So that the user does not have to remember
RS01	3	S	Option to save manually added stopwords	So that the user does not have to remember

Figure C1: User requirements (1/2). 'Cat' indicates the category, where 1 is business (application architecture), 2 is system (technical topic model implementation) and 3 is design (user interface and interactions). 'Prio' indicates the priority of the requirements according to the MoSCoW prioritization method, where 'M' stands for 'Must have', 'S' is 'Should have', 'C' is 'Could have' and 'W' is 'Will not have'.

RS11	3	S	Option to display the input data	To confirm the data is correct
RS12	3	S	Explanation and data specific information on how to change model parameters	The model parameters and its effects are hard to understand, so it should be clear what effect changes have on the data and topics
RS13	3	S	Option to evaluate model output based on individual documents	
RS14	3	S	Option to evaluate model output based on relationship between topics	
RS15	3	S	Visualizations that help understanding the topic distribution	
RS16	3	S	Include visualization on topic development over time	For preliminary data analysis within the platform before saving the topic output.
RS17	2, 3	S	The user should receive feedback after every interaction	
RS18	2, 3	S	Be informed on the input data quality and quantity	
RC1	2, 3	C	Warning if the input data quality or quantity is too low, negatively impacting the topic quality	The quality of the model is determined by the quality of the input; the user should be made aware of potential low quality
RC2	2, 3	C	Help on how to choose the optimal number of topics	It is not always clear to the user how to choose the number of topics
RC3	2, 3	C	Suggestions to model refinement	To support the user in refining topics with highest potential to quality improvement
RC4	3	C	The document-topic extraction should have one topic per data row	Other data analysis programs, like Tableau, are better able to handle big data tables with a large number of rows and small number of columns
RC5	3	C	Option to exclude all Yle news brand and department names	Exluding these stopwords at once saves the user time and effort.
RC6	3	C	Option to exclude all Finnish geographical names (cities, regions, etc)	Exluding these stopwords at once saves the user time and effort.
RW1	2	W	Option to delete a topic	When a topic does not make sense from user point of view. Not in final application because usage leads to decrease in output quality

Figure C2: User requirements (1/2). 'Cat' indicates the category, where 1 is business (application architecture), 2 is system (technical topic model implementation) and 3 is design (user interface and interactions). 'Prio' indicates the priority of the requirements according to the MoSCoW prioritization method, where 'M' stands for 'Must have', 'S' is 'Should have', 'C' is 'Could have' and 'W' is 'Will not have'.

D Design Choices

D.1 Comparison of topic modeling methods LDA and NMF

Intrinsic and extrinsic evaluation methods can be applied to measure the performance of the topic model. The quality of the model, whether the modeling was successful, can be measured by other evaluation methods. While it is hard to measure whether the extracted topics are 'good' in itself (intrinsic evaluation), measuring the usefulness of topics in broader use cases is more straight-forward (by extrinsic evaluation methods). Previous studies use different methods of evaluation, and results are rather inconsistent amongst the studies. Most conducted studies conclude that LDA shows better model output quality [55].

On the other hand, as pointed out before, NMF has better model consistency, lower empirical convergence time and user feedback is easier to incorporate in its non-probabilistic, matrix decomposition model. Choo et al. [11] compared the TM methods LDA and NMF quantitatively on three perspectives: consistency, empirical convergence and running time (see Table D1). Regarding consistency, they observed significant topic cluster membership changes with LDA (60-85%) whereas only relatively minor changes happened with NMF (10-25%). LDA's randomness nature can thus negatively influence the user experience of topic model interaction, since every time the model is run, a different result it obtained. Then, the randomness nature of the algorithm of LDA is also problematic for obtaining convergence. Even after 20,000 iterations, it was shown that 10-15% of the documents change cluster membership. NMF converges based on its own convergence criteria, leading to empirical convergence (no changes in document cluster membership) already at 60 iterations on average. Finally, because NMF reaches empirical convergence much faster than LDA (if at all), the running time of NMF is also much lower. In the same study, Choo et al. [11] found that LDA takes about 10 minutes to reach 20,000 iterations (with still relative cluster membership changes of 10-15% with each iteration), NMF takes only 48 seconds until it convergences (no relative cluster membership changes of documents). The running time of LDA linearly decreases by setting a smaller number of maximum iterations, but this still results in a longer running time with higher inconsistency compared to NMF.

Table D1: Comparison of the TM methods LDA and NMF by Choo et al. [11] regarding model consistency and convergence.

Measure	**LDA**	**NMF**
Consistency	60-85% change	10-20% change
Convergence (iterations)	10-15% topic membership change after 20,000 iterations	Converged after 60 iterations
Convergence (time)	10 minutes for 20,000 iterations	<1 minute

D.2 Tf-idf

Tf-idf is a numerical statistic that measures how prevalent terms are throughout the corpus. This weighting mechanism thus intends to reflect how important a word is to a document in a corpus. Term frequency (tf) indicates the significance of a term within a document, by taking the number of times this term occurs in a document with respect to the total number of terms in the document, expressing the probability of finding this word in a document. Inverse document frequency (idf) indicates how much information a specific words provides, i.e. how common or rare a word is in the set of documents. Words that occur only rarely in the corpus have a high idf value. This weight is calculated by logarithmically scaling the inverse fraction of documents containing the particular word:

$$\mathrm{idf}_i = \log\left(\frac{N}{\mathrm{df}_i}\right) \tag{D1}$$

where i is a term, df_i is the number of documents with i, and N is the total number of documents.

Multiplying tf and idf gives tf-idf (where i indicates a term and d a document)

$$\text{tf-idf}_{i,d} = \mathrm{tf}_{i,d} \cdot \mathrm{idf}_i \tag{D2}$$

A high tf-idf of a word is reached by a high term frequency in the document (td) and a low document frequency of the term in the whole corpus (idf). Applying tf-idf on the extracted BOW of the documents in a corpus normalizes terms, by weighing down the frequent terms and scaling up the rare terms.

The tf-idf score of words can be used to eliminate the most frequent and rare words from the produced BOW. Removing frequent and rare words influences topic modeling results. In the ITM application, users will be able to control minimum and maximum thresholds of document frequency of words. These thresholds will have a default value at model initialization, so that user control is enabled but not required.

D.3 Data input selection

Must-have requirements influencing the interface design of the data input selection that were extracted from the initial user study are (see Appendix C):

1. The option to choose the data source.
2. Option to filter the data source by metadata.
3. Option to choose the time window of the data source

The metadata to filter on depends on the data source. To not overload the user with (unavailable) options, the available metadata to filter on is best presented dynamically: only showing applicable filters.

During participatory design sessions, four additional (should-have and could-have) requirements were introduced:

1. Option to exclude individual documents.
2. Option to display the input data, to confirm whether the input data is as expected.
3. Displaying information on the quality and quantity of the input data, to estimate the model's performance.
4. Warning if the input data quality or quantity is too low, negatively impacting the topic quality.

The latter three requirements all concern feedback to the user, which separates them from the other requirements. According to the requirements, all feedback should be presented real-time (updated after every user or system action).

Participants mentioned that the data input does not require graphical visualization; a textual representation of the data and feedback provides sufficient information to start the modeling process. From a theoretical point of view, this also makes sense. We want the user to focus on the modeling process itself, since the topic modeling will be applied in bigger perspective than the input data. Thus, a table consisting of the document texts and metadata would suffice. The number of data items will be shown. Additionally, the distribution of text length (in number of words) of the documents will be presented, which can be used to estimate the quality of the data. As desired, a warning will be shown if the input data quality or quantity is not high enough to get good modeling results. If there are less than 250 documents to model, this warning will pop up.

D.4 Model configuration

Model configuration can be described as the hyperparameters that the user tunes *before* running the topic model. This does not mean the user should not be able to change them after the model has ran. If the user wants to change one or more of the hyperparameters of the model in an attempt to improve the model performance, then this should be possible. For example, the user should be able to change k, the number of topics to extract at any time. But, in contrast to model refinement, changes in model configuration requires the topic model to rerun. That is why this modeling step is considered separate from model refinement.

The only features of the data that the model is trained on are the words in the documents or corpus and its relative importance weighted by tf-idf. Since this is the only data input for the model, its quality influences the modeling output a lot, especially words that occur most frequently in the documents. This theoretical background was discussed in the initial user study, and decided was to give the users the opportunity to manually remove words from the corpus. Users indicated that visual aid was desired for deciding words to remove. This visual aid should reflect which words have a big influence on the topic model. Simply showing the most common words in the preprocessed corpus in a bar chart would suffice. In addition, concept testing rounds, in which the focus group participants actively interacted with

(parts of the) prototype ITM application, led to sets of use-case specific stopwords. For example, when performing topic modeling on a set of Finnish news articles, Yle-specific brand words like publishing departments and names of journalists, as well as Finnish geographical names like cities and regions, highly influenced the topics. An additional could-have requirement emerged: having the option to exclude these common set of words with one click in the interface, containing words depending on the use case.

As discussed before, the user should be able to control the feature selection even further by controlling set of words in the training corpus. Tuning the document frequency (df) of terms impacts the words taking into the topic modeling feature set. Setting df, is one way of giving the use control of the word representation in the training corpus. The document frequency of a term indicates how common or rare this term is in the corpus in relation to the documents; an accessible measure to non-technical users. Thresholds for a minimum and maximum df could be tuned by the user, leaving out terms from the corpus that occur too often (corpus-specific stop words) or too infrequently. Initially, no visual aid was offered to set these parameter values. The first prototypes consisted of sliders for setting the minimum and maximum df values, with explanation of the concept. But after user interaction observation, it became clear that the influence of these parameters on the model remained vague to the users. Thus, a table showing the the document frequency of each word is included in the application. This gives a clear picture to the users which words will be filtered out by setting df values, so that the influence of this parameter is no longer unclear.

Tuning the df paramater influences the tf-idf feature selection, and the topic model only indirectly. The topic modeling method itself, NMF, requires only one hyperparameter to be set by the user, which is k, the number of desired topics to extract from the data. As mentioned before, the users indicated that small as well as large numbers of k are desired. Since discovering large number of topics can only be successful if the underlying dataset is large enough, some measure, warning or limit should be implemented to prevent a bad and unexpected output. Chosen is to simply warn the user when selecting a large number of k, while having a small dataset.

From both literature and the user study we know that choosing the optimal number of k can be challenging. Choosing too few topics will lead to very broad results, while choosing too many topics will produce many small, similar topics. The user does not know beforehand how many topics would represent the data best, but this parameter should still be selected before running the model. Therefore, help on choosing k is provided in the application. There exists a number of model quality measures, which estimates the quality of the output of the model. One problem, however, arises with all of these measures: no measure is perfect in giving the quality of the model, because the quality of the model is best to be decided by a human, as mentioned by [31]. In this application, the quality of the model output with different numbers of k is estimated by in a quantitative way. Here, the *topic coherence* metric is applied to estimate the quality of the model output. This metric measures the coherence between the top words describing the topic, thus how semantically close

these words are. For model evaluation, measuring the topic coherence was chosen over other methods like residuals for multiple reasons. First, the main reason for choosing topic coherence over other measures is that this measures the model output without relating to its input. The goal of this topic modeling application is to learn topics that can be used in wider perspective, so the topics should not only represent the underlying training set (i.e. reasonable document-topic memberships), but should make sense from a human perspective. Error measurement methods like residuals, in which the difference between the predicted and actual values of the training set is measured, are based on how well the model represents the underlying dataset. Topic coherence, on the other hand, is measured by the model output (in this case the words that most frequently occur per topic). Thus, this measure describes the topics rather than the dataset.

A second reason is that experiments of taking the average topic residuals did not lead to any useful information for the user. The average residual decreases linearly with increasing k. This is because if a dataset is described by more topics, the topics become smaller and more specific, thus covering less documents in the dataset. The calculated difference between the actual words in the training set of documents and the predicted values by the topic model will be smaller, because the underlying dataset can be better described by the model. This behavior is also expected, and experienced when measuring the topic coherence, but to a lesser extent.

Moreover, topic coherence is preferred over the intrinsic evaluation metric *perplexity*. Perplexity (held-out likelihood) is widely used as evaluation method of language models, but Chang et al. [10] found that predictive likelihood measures and human judgement are often not correlated. Optimizing for perplexity may not yield to topics that are best human interpretable.

Finally, because topic modeling involves a mixed-membership model, we cannot use interactive visualizations which involves user judgement, for example dendrograms. Dendrograms visualize the hierarchical relationship between clusters, which may help humans understand how topics are formed and thus helps in deciding an optimal value of k. However, we cannot use dendrograms in mixed-membership models, since the clustering is not performed hierarchically and documents can belong to multiple clusters.

Then, there are different methods of calculating the coherence score. The two most popular methods are C_v, C_{NPMI} and C_{umass}. Lau et al. [29] found that, out of these and more topic coherence measures, the C_{NPMI} measure correlates the most with qualitative human judgement. More recently, O'Callaghan et al. [41] introduced a new way of calculating topic coherence scores achieving higher correlations with human judgement: *Topic Coherence-Word2Vec* ($TC\text{-}W2V$). This measure evaluates the relatedness of a set of top terms describing a topic, based on the similarity of their representations in a Word2Vec distributional semantic space. The coherence of a topic t_h represented by t top ranked terms is calculated by the mean pairwise cosine similarity between all relevant term vectors in the Word2Vec space by (where

the similarity is 1 if the distance between words in the space is 0):

$$\text{coh}\left(t_h\right) = \frac{1}{\binom{t}{2}} \sum_{j=2}^{t} \sum_{i=1}^{j-1} \cos\left(wv_i, wv_j\right) \tag{D3}$$

in which t_h is a topic and wv is de vector representation of a word.

The score for a topic model T with k topics is given by the mean of the individual topic coherence scores:

$$\text{coh}(T) = \frac{1}{k} \sum_{h=1}^{k} \text{coh}\left(t_h\right) \tag{D4}$$

A Word2Vec model will be trained on the collection of input documents, in which the words will be organized in a n-dimensional space according to their semantic similarity. The NMF model is trained for different values of k, and returns a *TC-W2V* value of each model configuration. The model with the highest estimated performance will not be automatically be chosen for the user. Instead, the user is presented with all the evaluation values of models with different k values in a plot. This way, we let the user decide on the desired number of k, instead of letting the model decide based on a quantitative measure. This also follows the design principle of giving the user the option to give input based on calculated information or suggestions.

D.5 Model output visualization

Model refinement can only be performed after analysing the initial topic model output. As discussed before, there are various methods of how the output of the model can be presented to the user. Overall, two topic representation methods can be distinguished: topic-term perspective and document-topic perspective. Both methods are considered in this application, because it yields different observations.

Topic-term perspective

Output representation from a topic-term perspective is important to consider in this context; the user is going to use the extracted topics outside of the scope of the current set of documents.

In collaboration with a group of end-users, decided is to include two visualization from topic-term perspective. In order to quickly judge on the generated topics, a list of most frequent words per topic will be given. Corresponding to results of a study done by Smith et al. [54], simple word lists, ordered by frequency within the topic, provide information efficiently. They suggested that a small number of words (10) per topic is good for quickly learning about the semantics of the topic, but that more words (they propose 100) are needed for better understanding. Because participants of our own user study indicated that 100 words per topic is too much information, and considering the possibly small size of dataset, the number of words representing each topic will be kept to 20. We found that representing just 10 topics does indeed not fully capture the meaning of some extracted topics.

Users indicated that this method is good for eyeballing (making estimates on model performance by looking at tables of 'raw' data without graphical representation). Because simple word list do not capture word and topic significance and relationships between topics and terms in topics and the corpus, a second visualization will be provided. This visualization is more complex and takes longer to evaluated, but reveal these additional information. Found was that, although its complexity, a visualization similar to LDAVis (Figure 9), satisfies. This type of visualization shows how much the topics are represented in the current corpus, how they relate to each other (by dimensionality reduction to two axes), and how top terms in the topics relate to top terms in the corpus.

Topics are automatically labeled by their topic number (ordered by topic size) and the three top words. This label provides sufficient information to indicate what the topic is about, without requiring any user input. These automatically generated labels should, however, be modifiable by the user. This way, the user is supported as much as possible, but is given the option to give input. This gives the user a feeling of control, encourages the user to critically review the cluster, and thus may improve the quality and interpretability of the cluster (see Model refinement in Section D.6).

Document-topic perspective

In context of this ITM application, topic modeling is performed to not only learn what topics are reflected by a set of documents, but mainly to apply them in a wider context (extracting topics from new documents and combining learned topics with other data sources). The main purpose of the application is to *discover topics* for later use, and not *clustering* the current document set. Thus, visualizations focused on informing the user about extracted topics are in this context seen as more valuable than visualizations representing how these topics are present in current individual documents. Nevertheless, document-term visualizations can still give meaningful information to the user to determine whether the model has extracted topics that make sense. The focus of document-term visualizations applied here is thus primarily on enabling validation.

The most simple document-topic representation would be displaying the extracted document-topic matrix from the NMF model. Here, documents are represented as rows, and columns indicate how much each topic is reflected in a document. The topic values for each document in the original output document-topic matrix W do not sum up to one, since NMF is invariant with column scaling of W and row scaling of topic-term matrix H. For interpretability of the results, column normalization is often applied to the document-topic matrix. Here, columns will be normalized to sum to unity by l_1-normalization ($\sum_j W_{ij} = 1$).

Similar to the topic-term perspective, a single matrix of results here does not represent the relation between the individual documents and topics. The relationship between individual documents how topics are represented in these documents can be represented in a two-dimensional plot, in which single data points are single documents. Using a dimensionality reduction method like PCA, t-SNE or UMAP, the high dimensional feature space of words can be scaled down (by approximation)

to two dimensions, trying to keep the relative distance between data points in the feature space best. When documents are represented in a two-dimensional feature space, users can compare the how related the documents are. To support the user in evaluating the topics suggested by the model, the document data points can be colored according to the topic which they most belong to. A downside of this visualization method is that involves assigning documents to one topic, although in reality multiple topics are reflected in a document. Another potential downside is that scaling the high dimensional features down to two axes might be misleading and confusing to the user, which might lead to misinterpretation of results. However, presenting an interactive visualization like this to participants in the focus groups led to positive reactions. The participants indicated that this type of visualization, although it displays results from document perspective instead of topic perspective, helps them in understanding the topic modeling results. Especially when the visualization is interactive, giving them the opportunity to explore individual clusters and following links to content and metadata of individual documents, this visualization was indicated as desired in the application.

The algorithms t-SNE and UMAP are the most common for non-linear, graph-based dimensionality reduction methods. UMAP (Uniform Manifold Approximation and Projection) [35] is competitive with T-distributed stochastic neighborhood embedding (t-SNE) [34] for visualization quality, and preserves more of the global structure, with significantly lower computation time. This increased performance in visualization quality as well as run time was also experienced in a small experiment. Both t-SNE and UMAP were implemented in two different prototypes and tested with similar NMF modeling outputs. The better visualization (important for interpretation by the user) and the lower run time (important for the user experience of the application) of UMAP led to preference of this method for the implementation of this visualization.

Because this visualization does not allow comparison of documents using the topic distribution, a third visualization is introduced. The user will be able to compare the (normalized) topic distribution of selected documents, by a stacked bar chart visualization. This supports the user in estimating whether the topic modeling output is reasonable by evaluating how topics are represented in single content. For example, when the user does not agree with the (degree of) topics assigned to a single document, the user is able to refine the model.

Comparing topic quality

In the model output, some topics may represent the documents well, while other topics may not because they are too general. But as mentioned before in literature, due to its latent nature, the output of topic modeling is highly subjective. No perfect measure was found in previous studies, because we deal with human subjectivity. On the other hand, comparing the quality of individual topics in a model (which is different from comparing the quality of different topic models), provides useful information to the user about the modeling results. Topics of low quality are more likely to be improved when users refine those topics. Moreover, attempting to refine

topics of high quality sometimes leads to decreasing the overall quality [31]. It is thus questionable whether the user experience and the final ITM output will be improved by presenting a statistical quality measure of individual topics in the interface. From the user studies we learned that users sometimes find it hard to decide which topics to focus on in the refinement process. Therefore, decided is that a topic quality measure, suggesting which topics are worst and might benefit from manipulation, will be presented to the user. In contrast to the topic coherence measure presented before, used to estimate the overall model performance, it was found that an error measurement (sum of residuals) is preferred over topic coherence to estimate individual topic quality. A small experiment was conducted, in which the residuals of individual topics of a model output were compared to the topic coherence of the same topics. The residuals for each topic and document indicate how well a topic approximates the underlying dataset. Residuals are differences between observed and predicted values of data. The lower the residual, the better the topic approximates the text of a document. The residual of the topics can calculated by the Frobenius norm of the original document-term matrix (matrix A) minus the dot product of the coefficients of the calculated topics (matrix W) and the topics (matrix H). The average residual for each topic is then the measure shown to the user, which is a number that can be compared over different topics to find the topic with the highest and lowest residual on average. Domain experts indicated that they agreed more with the quality measure of the residuals than the topic coherence. This will thus also be implemented in the ITM application. This quality measure should be presented to the user along with an explanation of how to interpret this, so that is clear that this is just an estimation and the user will be aware of the fact that their own interpretation of topics should lead every decision. Users should still be encouraged to use their own empirical review.

On the other hand, decided is that no statistical measure of the *overall* model performance will be presented to the user. Because the user is usually only using the ITM application to derive one set of topics from a single set of documents, there will be no baseline to compare the modeling output to. So, next to human subjectivity of the latent output, this is a reason not to implement a global model performance measure.

D.6 Model refinement

The fourth step of the ITM application is where the user interaction with the actual model comes in, the part where the *human* is taken *in the modeling loop* after data selection and tuning the model configuration. The visualizations from topic and document perspective inform the user about the model performance. Note that there is no statistical measure of the overall model performance presented to the user. Besides the residual measure to compare the quality of individual topics as indication which topics to refine, users of this ITM application are left to their own empirical review. Using the various visualizations given in the interface, the users are able to explore the extracted topics and their relations, and how they are represented in the documents. This should give the user sufficient insight on the latent topics and

how they represent the data. Based on this, the user should able to make changes to either the model configuration or its output. To fine-tune the model settings, the user should be able to go back to the previous step. Here, preprocessing can be adjusted (e.g. adding stopwords if a word is found to influence the topics too much) and the model's only hyperparameter, the number of topics k, can be changed. Making changes to the model configuration requires a re-initialization of the model. Model refinement, however, concerns with model revision interactions which happen after the initial model output. From previous ITM applications we know a lot of revision techniques (see Table 4). These techniques were also proposed and discussed in the focus groups. As noted before, all potential end-users indicated to prefer a simple application, in which not too many user actions are needed to get results. In addition, the users indicated that only high-level changes to the model would suffice. Very low-level revision interactions, like changing the weights of keywords in a topic, are not desired.

Finally, users indicated that they would like to have the possibility to **merge**, **split** and **delete** (topic level), and to **remove keywords from a topic** (word level). All four methods were implemented, but after initial evaluation on the methods, the **delete** option was removed. Observed was that users tend to remove a topic from the model output if they do not fully understand the topic, for example because the top terms in the topics are not very informative. After researching the topics that the users wanted to remove by conducting individual documents scoring high on these topics, it became clear that all documents per topic were actually related and could be described by some latent theme. So, instead of deleting a topic, the user should be encouraged to look at documents in which this topics is represented high, to find out the latent theme.

The results of any refinement done by the user should be shown immediately. The users would like to see suggestions on which topics to refine, which is enables by presenting the topic quality estimations (residuals).

Manual topic labeling

As mentioned, the topics will be automatically labeled by the topic number (ordered by topic size) and the three most prevalent words. The user is encouraged to change these names, because the three top words usually do not capture the meaning of the topic best. Renaming topics requires in-depth human evaluation of the topics. This is why naming the topics is not mandatory, it would require a lot of human effort to name all the topics, especially in the very likely case of further model refinement. But for the same reason, topic renaming is still recommended. When users are evaluating the individual topics to come up with good descriptors, they are, perhaps unconsciously, evaluating the topic itself. When inconsistencies are found, the users are still able to refine the topic. It is thus good practice to let the user evaluate and name a topic. For these reasons, and keeping the design principles in mind, topic renaming will therefore be *optional* in the application. A suggestion is made by the application, which support the user, but user input is possible.

D.7 Exploratory data analysis of the output in wider context

The last part of the interactive topic modeling application involves exploratory data analysis (EDA) of the modeling results on the training data as well as other data. This part is not supported in all ITM applications introduced earlier, but some initial EDA is desired by the stakeholders in this use case. Users indicated that the main goal of using the topic modeling application would be to explore and discover which topics and themes of the provided content are interesting for further analysis. They want to find out which topics are different from other topics, and how and why these topics stand out. In collaboration with the stakeholders in the focus group sessions, we came up with two ways of enabling initial EDA: topic development over time and comparing topic content production and consumption.

Topic development over time

How topics develop over time gives an initial idea on how the topic model output can be used to discover trends in the produced media. Topic development can be represented by a line graph and a stacked bar chart with the produced content (y-axis) over time (x-axis). Line charts better visualize low number of k, while bar charts become more useful compared to line charts that tend to get messy when k becomes larger.

The produced content per topic should be the sum of the normalized values of each document per topic, since each topic is usually only represented by single documents to a certain extend. Trends can be discovered by analysing sudden changes of produced content per topic in the graph. Additionally, the size of the size of topics can be read from the graphs, which gives the user the opportunity to compare the importance of each topics not only over time, but also relative to other topics.

Comparing content production and consumption per topic

Secondly, from the user studies we know that the main interest of media data analysis is media planning according to goals defined by other stakeholders. Comparing audience data with the produced media might lead to insights which produced media is preferred over other by audience groups. If the media is clustered into a number of topics, then comparing production and consumption data can inform which topics are more popular than others. Stakeholders suggested that comparing consumption data over different target groups (defined by age), with the production data over topics is their major interest, to learn about any differences between media watching and reading behavior in different themes. Insights could lead to refocus or directing media planning towards or away from certain topics for different target groups.

A visualization was proposed to the end-users, in which the relative production (first y-axis) and consumption of different target groups (second y-axis) are plotted over the topics (x-axis).

To enable this EDA, consumption data should be imported into the application. The production data here is the training data of the model, consisting of the individual documents, distributed over the extracted topics. The relative production (y-axis) per topic (x-axis) is calculated by summing per topic how much it is represented (normalized per document) in each document, and taking this value as a percentage over the total of these topic production sums, see Formula D5.

$$\text{relative\_production}(t) = \frac{\sum_{d=0}^{m} w_{dt}}{m} \cdot 100\% \tag{D5}$$

where $w_{dt} \in \mathbf{W} \in \mathbb{R}^{m \cdot k}$, k is the total number of topics, d is a single document, t is a single topic, m is the total number of documents, and $\mathbf{W} \in \mathbb{R}^{m \cdot k}$ is the document-topic matrix.

The consumption data can be calculated similar manner, see Formula D6. The consumption of an individual document is represented by the minutes that this piece of content is read (articles), watched (video) or listened (audio). Whereas the production of all documents is equal, the consumption rates thus differ per document. When a consumption value of a piece of content is multiplied with the value of how much this document belongs to one topic, we get the consumption value of this topic of one document. We can do this for all topics, and for all documents, to get the consumption values of all documents over the topics. When these values of the topics are also taken as percentages of the total, they can be compared with the production per topic percentages. The consumption data can be easily divided into the different target groups, by performing the previous step for every desired target group, in which the consumption data (in minutes per document) differs per target group.

$$\text{relative\_consumption}(a, t) = \frac{\sum_{d=0}^{m}(w_{dt} \cdot \text{consumption}(d, a))}{\sum_{d=0}^{m}(m \cdot \text{consumption}(d, a))} \cdot 100\% \tag{D6}$$

where a is the agegroup, $w_{dt} \in \mathbf{W} \in \mathbb{R}^{m \cdot k}$, k is the total number of topics, d is a single document, t is a single topic, m is the total number of documents, $\mathbf{W} \in \mathbb{R}^{m \cdot k}$ is the document-topic matrix, and $\text{consumption}(d, a)$ is defined by

$$\text{consumption}(d, a) = \frac{c_{da}}{\sum_{d=0}^{m} c_{da}} \tag{D7}$$

where $c_{da} \in \mathbf{C} \in \mathbb{R}^{m \cdot a}$, and $\mathbf{C}$ is a matrix containing the consumption in total reading minutes per agegroup a per article d.

Data export

To allow for further data analysis and application of the derived topics, the data used in this application should be made available for export. The user should be able to download the 'raw' model output (the topic-term matrix and the document-topic matrix) and the training data as well as consumption data. In addition, an option will be made available to download a file consisting of all the model settings and modeling steps performed by the user. This requirement came from one of the focus

group sessions, and provides explanation to the modeler, as well as other stakeholders, how the topics are created from the dataset. All of these data outputs will be in CSV, since this is the most common data file type used for data analysis by the stakeholders.

Moreover, it is preferred that all visualizations are easily downloadable, so that they can be used directly into data reports or presentations. This can be enabled by a download button per visualization. The interactive nature of visualizations can be kept by providing a HTML download, other static visualizations should be downloaded in PNG format.

D.8 Design Heuristics

The 10 design heuristics according to Jakob Nielsen [38] and how they are applied in the design of the ITM application.

1. **Visibility of system status**. The ITM application will show when the model is running, so that the users will be informed at all times when they can interact with the system. In addition, all visualizations reflect the most recent model output, so the status of the model can be read at all times.

2. **Match between system and the real world**. The words, phrases and concepts used in the interface will be in the users' language. Users are domain experts without data science background, so natural, real-world conventions will be followed instead of system-oriented and too technical terms.

3. **User control and freedom**. *Undo* and *redo* options would improve the usability of the system, but will not be implemented in the first production version application researched in this thesis because of its complexity of system status and calculations. However, it is recommended to implement this in the future, see Section 5.2.

4. **Consistency and standards**. Platform convention are followed as much as possible. Users know what actions buttons, checkboxes, text fields etc. require and result, so these general interaction concepts are used. In addition to design consistency, model consistency is also considered. The topic model gives consistent result, to not confuse the user, gain trust and give the user a sense of control.

5. **Error prevention**. Warning messages are presented when a user action, data selection or model setting might result to bad or unexpected results. User mistakes will be avoided as much as possible by asking for confirmation before applying changes to the model.

6. **Recognition rather than recall**. The user's memory load will be minimized by making objects, actions and options visible. The user will be able to output model settings and all interactions performed at the end of the modeling process, so that this does not have to be remembered for future modeling scenarios.

7. **Flexibility and efficiency of use**. Not all features and visualizations will be rendered by default in the application. Rather, users are given the option to render additional visualizations and performance calculations. Parameter tuning is also optional, so that novice users are not required to know the modeling process in depth.

8. **Aesthetic and minimalistic design**. It will be sought to create a minimalistic but aesthetically pleasing design, containing only relevant information at different modeling stages.

9. **Help users recognize, diagnose and recover from errors**. Error and warning messages will be shown in natural language as much as possible, so that the user will recognize issues.

10. **Help and documentation**. Explanation on how to use the application, as well as best practices to obtain good modeling results, will be included. Contact information in case of further questions or issues will also be included.

E Implementation diagrams

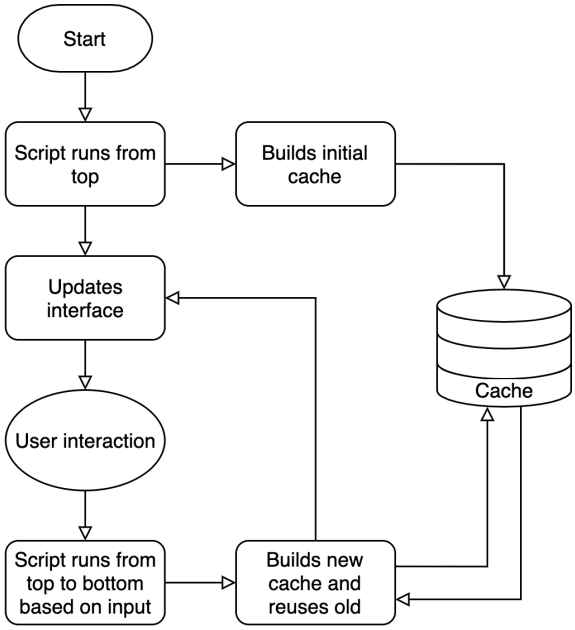

Figure E1: Streamlit flow. Every time the user interacts with the application, the script is run from top to bottom to update the UI. Cache is stored and reused for UI generation to optimize performance.

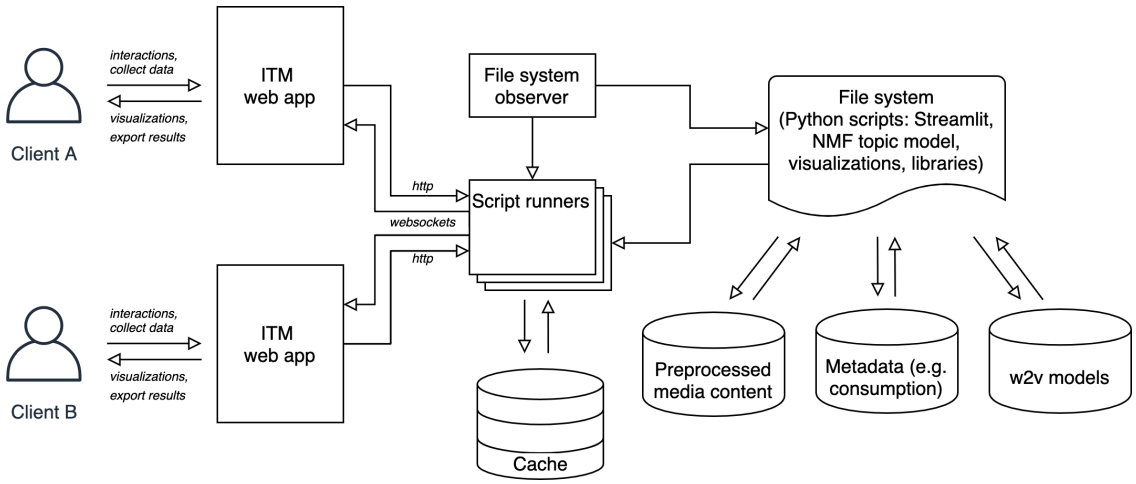

Figure E2: Back-end architecture. Multiple clients can use the application simultaneously, managed by script runners and a file system observer.

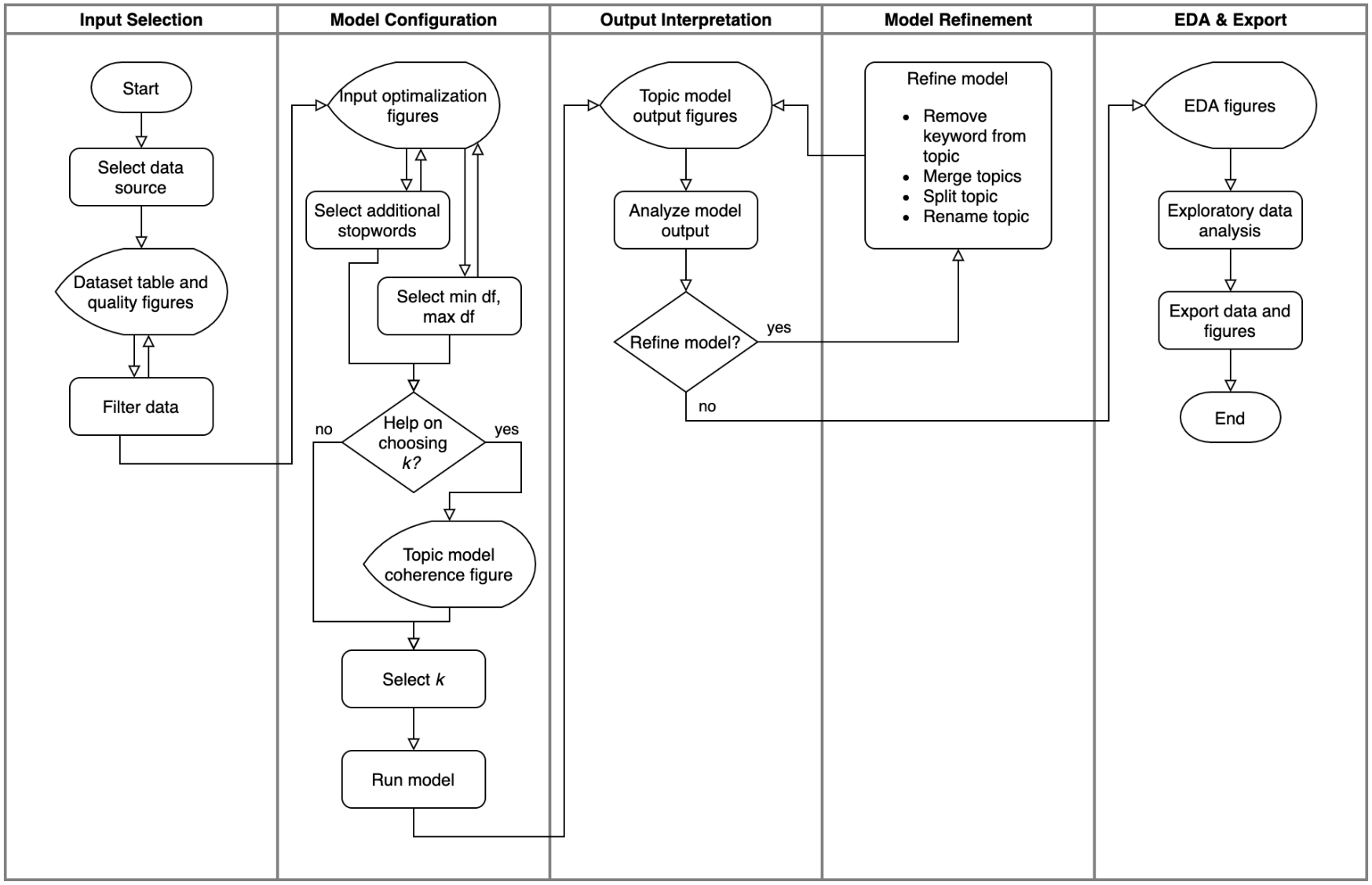

Figure E3: User flow diagram. There user flow is divided in five steps, but the user is free to jump back at any time to make changes to the model.

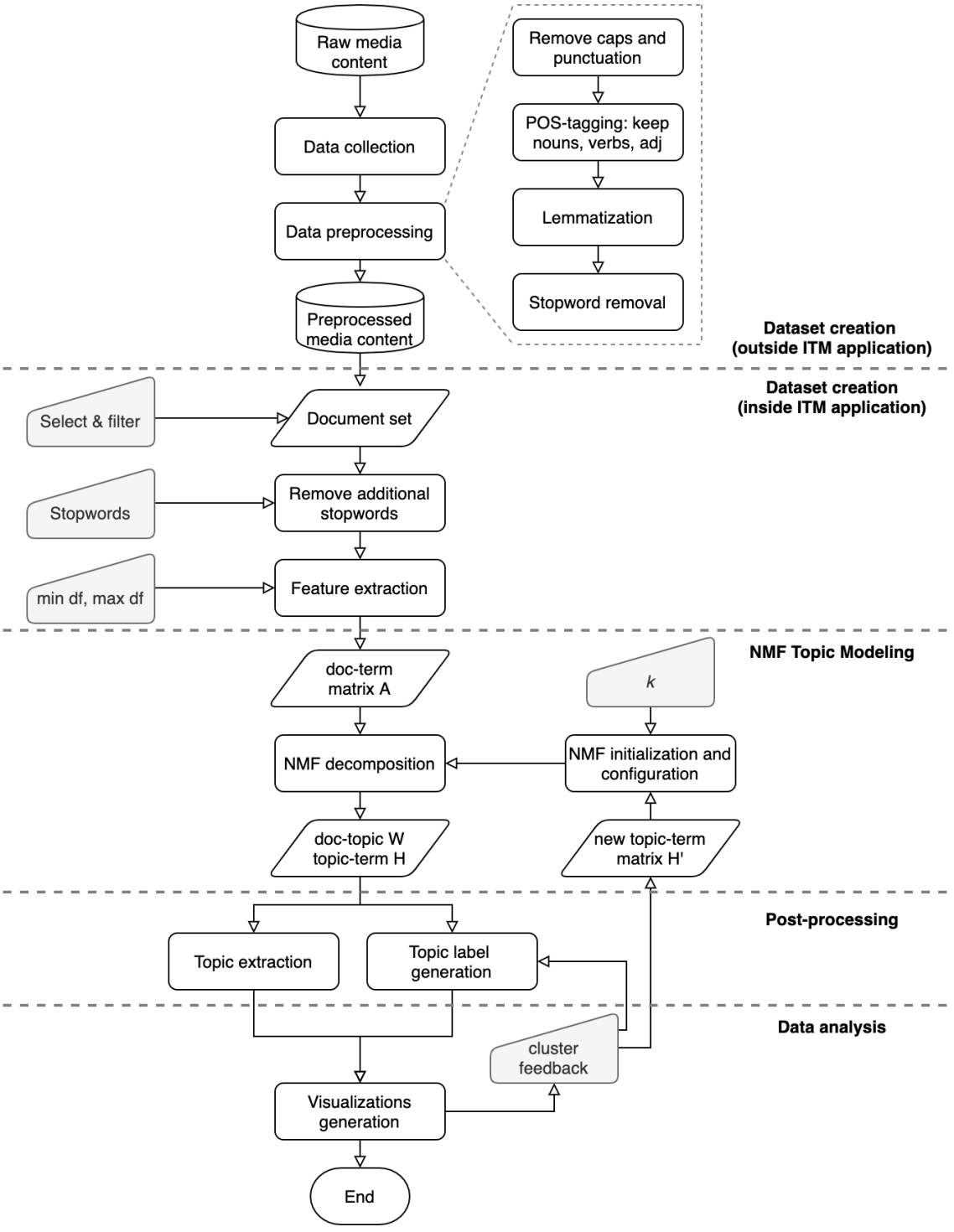

Figure E4: Interactive Topic Model workflow. Most of the preprocessing happens outside the ITM application; only manual changes by the user to the dataset happen in the ITM application. Light gray manual input boxes indicate user input.

F Yle specific stopwords

Default stopwords used in the evaluation study are (in Finnish, explanation or English translation in brackets):

‘uutiset’	(‘news’)
‘Yle’	
‘A studio’	(a TV and Radio publishing department within Yle)
‘Sannikka Ukkola’	(a current affairs program by Yle hosted on Yle TV1)
‘8 minuuttia’	(a type of news program within Yle)
‘Yle perjantai’	(a type of news program with Yle)
‘A talk’	(a show hosted by A-studio)
‘MOT’	(TV show of Yle)
‘puoli seitsemän’	(news TV show of Yle)’
‘politiikka’	(‘politics’)
‘kotimaan’	(‘domestic’)
‘ulkomaan’	(‘foreign’)
‘kotimaan uutiset’	(‘domestic news’)
‘ulkomaat’	(‘foreign countries’)
‘ulkomaan uutiset’	(‘foreign news’)
‘talous’	(‘economy’)
‘terveys’	(‘health’)
‘elämäntapa’	(‘lifestyle’)
‘kulttuuri’	(‘culture’)
‘suomi’	(‘Finland’)
‘yleiset’	(‘general’)
‘Yle uutiset 60 vuotta’	(‘60 years of Yle news’ a Yle show of historic news).

G Evaluation study questionnaires

G.1 User background questionnaire

1. Which category below includes your age? (17 or younger, 18-20, 21-29, 30-39, 40-49, 50-59, 60 or older)
2. What is your gender? (Male/Female/Prefer not to answer)
3. I am confident in my ability to use data analytics tools
4. I am confident in my ability to analyse data
5. I am confident in my ability to use data to support decision making
6. I am confident in my ability to use models for data prediction and machine learning
7. I am confident in my ability to create models for data prediction and machine learning
8. I have used models for topic modeling or clustering before
9. I have made models for topic modeling or clustering before

Questions 2 to 7 about data and analytics skill assessment are adopted from the Open University Library Data project [40].

G.2 SUS questionnaire

1. I think that I would like to use the application frequently.
2. I found the application unnecessarily complex.
3. I thought the application was easy to use.
4. I think that I would need the support of a technical person to be able to use this application.
5. I found the various functions in this application were well integrated.
6. I thought there was too much inconsistency in this application.
7. I would imagine that most people would learn to use this application very quickly.
8. I found the application very cumbersome to use.
9. I felt very confident using the application.
10. I needed to learn a lot of things before I could get going with this application.

G.3 User experience and user perception questionnaire

1. Using this application to perform the task was frustrating.
2. I trusted that the application would update the clusters of the articles well.
3. It was easy to use this application to perform the task.
4. I was confident in my specified changes to the tool.
5. How satisfied are you with the final topics?
6. How do you think the final topics compare to the initial suggested topics?
7. After my changes, the application updated fast enough.
8. The tool made the changes I asked it to make.
9. The tool made unexpected changes beyond what I asked to make.

G.4 Post-task semi-structured interview questions

The questions are highly dependent on the modeling process and spoken thoughts of the participants during the process.

- Why did you choose for this number of topics?
- What do you think about the on-page explanations?
- Did the visualizations provide you with sufficient information?
- Was every function and visualization clear?
- Does this application help you in discovering new insights or new use cases?
- Are there functions, visualizations or data that where not implemented but you would like to use?