Progressive Learning of Topic Modeling Parameters: A Visual Analytics Framework

Mennatallah El-Assady[1,2], Rita Sevastjanova[1], Fabian Sperrle[1], Daniel Keim[1], and Christopher Collins[2]

[1]University of Konstanz, Germany
[2]University of Ontario Institute of Technology, Canada

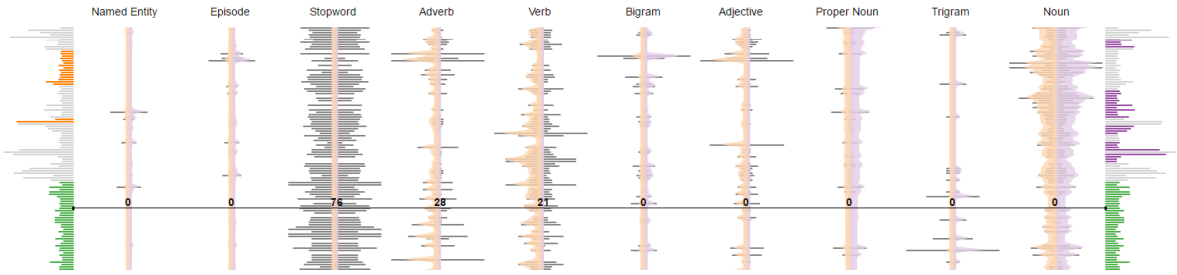


Fig. 1. *Parameter Distribution View* using comparative bar charts. This compact visualization technique enhances the comparison of two parameter distributions using mirrored bar-charts as a baseline and two asymmetrical violin-style plots as distribution estimates. The plots are scaled using the ratio between the two compared assortments (on both sides). The larger value is scaled to the full width of the baseline and the smaller value is scaled proportionally. This figure depicts the comparison of the utterance descriptor features of the second US presidential debate between Obama and Romney in 2012. All utterances are sorted according to their topic coherence.

Abstract— Topic modeling algorithms are widely used to analyze the thematic composition of text corpora but remain difficult to interpret and adjust. Addressing these limitations, we present a modular visual analytics framework, tackling the understandability and adaptability of topic models through a user-driven reinforcement learning process which does not require a deep understanding of the underlying topic modeling algorithms. Given a document corpus, our approach initializes two algorithm configurations based on a parameter space analysis that enhances document separability. We abstract the model complexity in an interactive visual workspace for exploring the automatic matching results of two models, investigating topic summaries, analyzing parameter distributions, and reviewing documents. The main contribution of our work is an iterative decision-making technique in which users provide a document-based relevance feedback that allows the framework to converge to a user-endorsed topic distribution. We also report feedback from a two-stage study which shows that our technique results in topic model quality improvements on two independent measures.

1 Introduction

Topic modeling algorithms are a class of unsupervised machine learning algorithms which categorize collections of documents based on the distribution of topics discovered within. They are often used to gain insight into the content of document collections without the need for time-consuming classification and close-reading. Topic models have also been widely used as processing steps in automatic text analysis and visualization approaches [23]. Despite their convenience and wide applicability, these models typically remain black-boxes, not readily understood by end users [11, 25]. However, understanding the basic principles of these algorithms is essential in order to properly configure and use them. Hence, there is a need to **understand** how the results of topic models are created and to **adapt** the models to given data and tasks, in order to enhance a model's provenance and reliability [4]. We created a technique that can provide understanding about topic models and an ability to adapt them to specific data and tasks, without requiring users to become proficient experts in the underlying code and settings.

Topic models are notoriously difficult to work with [7]. As the recent paper investigating how non-experts perceive, interpret, and fix topic models put it, "with an LDA-based approach [..], seemingly small changes on the user side could have unpredictable and nonsensical cascading side effects" [25]. Yet, Blei argues that their power will be realized best when used in the service of history, sociology, linguistics, and other social sciences and humanities fields [4]. This is usually accomplished through teaming computer scientists with non-computer scientists to create topic models together. However, with the popularity of toolkits such as MALLET [28], it is becoming more common for people at all levels of expertise to generate topic models. Consequently, it is critical to create a technique that enhances the understandability and adaptability of the parameters by non-experts.

Designing model-driven visualization approaches to enhance the interpretability and trust for automatic text analysis techniques has proven helpful [11]. Visual analytics enables data- and task-centric model creation through a human-in-the-loop design. Hence, an effective model visualization with an iterative feedback cycle is a promising approach for a user-steerable and interpretable topic modeling process. Such a process could be especially helpful for humanities and social science scholars to make use of large text corpora through quick processing.

The visual analytics process of our technique is shown in Fig. 2, combining automated parameter space analysis, topic matching, and summarization, with a visual analytics dashboard consisting of several linked views of competing modeling results and interaction techniques for users to provide feedback and adjust the models. Our goal is to address the problem of controlling the model without having to read all the documents (which takes time) or understand the mathematics behind the algorithms (which requires effort). We strive for intuitive types of feedback, mirroring those recommended by Lee et al. [25],

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

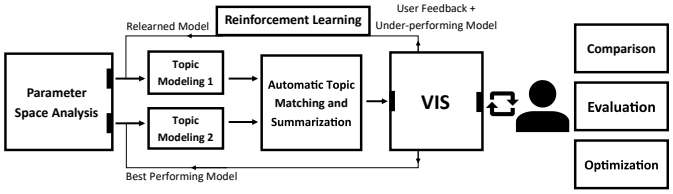


Fig. 2. The progressive learning process, including an initial parameter space analysis and an iterative human-in-the-loop reinforcement learning process in which human annotators compare, evaluate, and optimize models using a visual analytics dashboard.

such as *which topic better suits a document* or *which word does not belong in a keyword set*.

The user feedback is used to generate new candidate models which can be further refined. The process supports users in comparing, evaluating, and optimizing topic models in order to achieve an output which more coherently describes the document collection. The visualization step of the process is designed with four linked views, each to support a task: exploring the automatic matching results of two models (**Topic Matching**), investigating topic summaries (**Topic Summarization**), analyzing parameter distributions (**Parameter Distribution Analysis**), and reviewing documents (**Document Relevance Feedback**). Our tasks are inspired by the model understanding and comparison tasks proposed by Alexander and Gleicher [2]. However, where they choose to be *parameter agnostic* we choose to reveal the parameter space and how the values affect the model.

The amount of feedback to provide in each iteration is up to the user. The more documents rated, the closer the next model will be to the ideal topic composition. However, at some point the cost of the refinement process outweighs the benefits of an unsupervised algorithm. The balance would be providing minimum feedback for maximum improvement. Thus our visualizations are also targeted at guiding users to those ambiguous documents and topics for which feedback would have the most impact on the next iteration of learning. Through enhancing feature distributions and descriptor keywords we enable users to analyze the effects of parameters on topic models, understand the impact of document descriptors on the topic keyword vectors through topic summarization, and, ultimately, optimize the topic modeling results in an iterative loop using reinforcement learning.

We evaluated our technique with a mixed-methods study using empirical quality metrics alongside human-expert judgments [22]. Six participants carried out model refinement using our technique, and the results were analyzed both with quantitative metrics of topic coherence and uncertainty, as well as manual quality coding carried out by three experts in the domain of the data. All measures showed improvements in topic model quality after several learning iterations.

Our research makes the following contributions: (1) We introduce a human-in-the-loop progressive learning technique for topic model refinement, which is independent of the specific topic modeling approach. (2) We present four linked task-oriented visualizations for enhancing understanding of topic model parameters and providing intuitive feedback about model quality. (3) We validate our technique with both empirical and qualitative measures.

2 Background and Related Work

There are two main classes of topic models: probabilistic models, including the popular Latent Dirichlet Allocation (LDA) approach [6], and non-probabilistic approaches, most prominently, Non-negative Matrix Factorization (NMF) [43]. Probabilistic models (e.g., [5,30,33,39]) are the most prominent and are based on the assumption of the existence of a latent space [24] in which relations between objects are determined. Generally, probabilistic approaches can produce higher quality results, but at the price of determinism and stability on refinements. Both types of models have several factors in common, including *input parameters* which specify the model characteristics, such as the number of topics or weightings on classes of words in the input documents, *keyword vectors* which are ranked lists of words which represent extracted topics, and *document descriptors*, which are vectors of scores relating each document to each topic. A comprehensive survey of different probabilistic topic modeling approaches is provided by Blei [4].

In recent years, various interactive visualization approaches have been developed for the content analysis and exploration of document collections. Mostly, these are based on the LDA model [6]. Most approaches utilize ThemeRiver-based [21] visualizations to highlight temporal trends in topics, e.g., TextFlow [13], RoseRiver [14], Visual Backchannel [16], and TIARA [41]. Other approaches go beyond exploring the temporal dynamics, e.g., Paralleltopics [17], Hiérarchie [37], Hierarchicaltopics [18], UTOPIAN [8], Termite [10], and Serendip [3]. However, the most relevant visual analysis approach to our technique is the task-driven comparison of topic models by Alexander and Gleicher [2]. Using their so-called "Buddy plots", they highlight the differences in the modeling results between two different models by fixing one or multiple topics. In addition, this paper categorizes three topic modeling tasks: Understanding topics, understanding similarity, and understanding change. They perform the comparison of topic models using three techniques: topic alignment, distance comparison, and timeline comparison. While this paper paves the way for comparative topic analysis, our technique is designed to go beyond single comparisons and extends the analysis to iterative optimization cycles.

A more recent trend around the analysis of topic models is the enhancement of the comprehension and interpretation of their results. This is motivated by the evidence that most currently used automated, likelihood based quality measures for topic models do not capture their quality correctly. In fact, Chang et al. found that they are actually negatively correlated with the perceived quality [7]. To improve this situation we have to take interpretation and trust into account when designing models and tools, a fact that has often been overlooked in the past, as emphasized by Chuang et al. [9, 11]. They contributed a set of guidelines that should be employed when developing new models. Trustworthy and reproducible topic models are especially important for social sciences, where Ramage et al. [32] find there is strong and growing demand.

In order to make systems more understandable for non-experts, Lee et al. [25] isolated a few primitive interactions that were intuitive to non-experts, such as adding and removing keywords from topics. Choo et al. [8] propose to give users the option to add individual weights to single keywords in order to reach a more understandable topic model. Both of these propositions are present in our approach.

3 Parameter Space Analysis

Topic models typically operate in a vector space [34] defined by the accumulated keyword frequency vectors of all documents in the analyzed corpus. These document descriptor vectors are constructed using a bag-of-words model, which weights every keyword in the vector by its overall frequency in the document. These weights can be adjusted by parameters, which are initialized in a preprocessing step. Topic models work best if documents can be associated strongly with one topic, and the topics generated have minimal overlap. To achieve this, we need to find ways to make documents separable through appropriate parameterization. One common way to use parameters is to provide weights to classes of words (e.g., parts-of-speech). We call these classes of words the *features* and their parameters *feature weights*. For example, a feature weight could be used to downweight all function words (*stop words*), effectively removing them from consideration in the modeling. Similarly, a feature weight could be used to boost the impact of all proper nouns. Discovering what the appropriate choices of feature weights are is not well supported by topic modeling toolkits, and the values are often specific to a dataset and very sensitive to change.

In order start with feature weighting parameters appropriate for the data, we propose a two-step strategy, in compliance with the findings of Sedlmair et al. [35]. First, we compile an automatic "educated guess" for the data-driven feature selection and weighting, which can then be adjusted by the user. Second, we generate document descriptor vectors using a scoring function selected by the user. Through this process, we configure the initial run of the topic modeling algorithms in our progressive learning process, as shown in Fig. 2.

3.1 Data-Driven Feature Selection and Weighting

In the context of our parameter space analysis, the problem of data-driven feature selection and weighting is defined over an abstract set of disjunct features $\mathcal{F} = \{\vec{f}_1, \vec{f}_2, ..., \vec{f}_n\}$ across multiple documents $\mathcal{D} = \{d_1, d_2, ..., d_m\}$, with vector $\vec{f}_i = < w_{i,1}, w_{i,2}, ..., w_{i,v} >$ representing the complete signature vector of v words w comprising the feature. Every document is defined over the set of all features as $d_j = \{\vec{f}_{1,j}, \vec{f}_{2,j}, ..., \vec{f}_{n,j}\}$, with $\vec{f}_{i,j}$ as the concrete feature frequency vector of the document. For example, in our work, we currently consider the following set of features $\mathcal{F} = \{\vec{f}_{nouns}, \vec{f}_{verbs}, \vec{f}_{adverbs}, \vec{f}_{adjectives}, \vec{f}_{propernouns}, \vec{f}_{namedentities}, \vec{f}_{episodes}, \vec{f}_{bigrams}, \vec{f}_{trigrams}, \vec{f}_{stopwords}\}$. Here, $\vec{f}_{nouns}$ is a vector of all *nouns* in the corpus. Hence, in analogy to the bag-of-word representation for documents, we can define every document as a set of frequency vectors given our feature set $\mathcal{F}$. The feature collection we use is driven by norms in topic modeling, but is modular and can be extended to accommodate different tasks or properties of the documents.

In order to select the appropriate features for a given corpus and weight them, we analyze their *discriminativeness* which is defined by a non-uniform feature distribution across all documents. We compute this by first calculating the pairwise *feature variability* over all documents for every feature $\vec{f}_i$. We provide five different measures of feature variability in our framework. Our feature variability measures each take two concrete feature vectors $f_{i,j}$ and output a single scalar. The default measure is a *diversity index* defined by the feature entropy [15]. Our experiments with different document collections confirms the finding of Oelke et al. [31] that the entropy is a well suited measure to enhance document separability. In addition, we provide alternative measures, such as feature vector distances, e.g., cosine similarity and inverse document frequency, as well as a set overlap coefficient [26, 36]. The last alternative measure that is available is RWPD, a ranked and weighted penalty distance, which we introduce in Sect. 4.1.

Next, given the distribution of feature variability values across all document pairs for each feature f_i, we calculate the standard deviation σ_i of the distribution. Finally, the ratio of σ_i for every feature compared to the minimum σ_{min} across all features is proportional to the ratio of discrimination of these features for the given corpus [1]. The result is that features with more diversity of values across documents (i.e., those that are more discriminative) are scaled to larger values. These ratios thus become the initial feature weights. Based on these measures, our framework analyzes any given data set and computes a suggestion of discriminative features and their weighting. These suggestions can be used directly or interactively refined by the users.

For some datasets, the discriminativeness of features can overemphasize different aspects of the documents, e.g., the idiosyncratic use of language by different authors or speakers. That is, topics based on these feature weights would separate utterances by speaker rather than by content. This is a common problem that also affects out-of-the-box topic modeling algorithms [25]. In order to counteract the oversensitivity of the parameter space analysis towards linguistic nuances, such as writing styles, and to focus on a content-based separation, we introduce a globally learned parameter scale that can be weighted into the individual data-driven weights, as described in Sect. 5.3. This global score captures successful feature weight distributions for different text types, from large corpora. Depending on the analyzed text genre, such a normalization can be vital for the topic modeling quality.

3.2 Document Descriptor Vector Generation

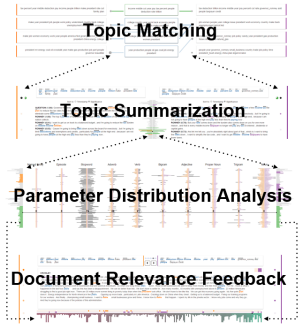
Starting with the computed feature weights from the first step of the parameter space analysis, we derive *document descriptor vectors* which assign each word of the document an *importance score*. First, we multiply the concrete feature frequency vectors by the feature weights to obtain a weighted feature vector for every document. These are the default document descriptor vectors, based on the word frequency. However, as Collins et al. state, frequency is not necessarily very effective at scoring key terms of documents [12]. Consequently, we allow users to select an alternative descriptor scoring function that replaces the frequency-based score in order to enhance the vectors' descriptiveness. The currently supported scoring functions include tf-idf [38] and ttf-idf, an adapted version of tf-idf using the total term frequency over all documents, log-likelihood ratio [27], and measures based on semantic similarity such as word2vec [29]. Where the feature weighting step weights features across the corpus (e.g., upweighting nouns), these scoring functions weight each word for each document (e.g., upweighting "taxes" as a key term for d_1). After being calculated and normalized to integer values, the document descriptors are then used as input for the topic modeling algorithms.

3.3 Initializing the Workspace

Our learning technique iteratively compares two topic models at a time. Therefore we create two initial parameter configurations through parameter space analysis to compute the two models. These configurations could be exactly the same (for example in order to examine non-deterministic topic modeling stability and robustness) or could consist of different feature distributions or weightings. Since our technique is independent of specific topic modeling algorithms, we allow the users to choose the two models (could be the same model twice) from a set of probabilistic and non-probabilistic models.

4 Visual Analysis Workspace

The core component of our visual analytics technique is the visual analysis workspace. This is the interface in which users interact with the processed data and topic modeling output. We designed the visual interface as a dynamic workspace with consistent visual encoding to facilitate performing the mentally challenging exercise of comparing the different models and their document distributions. One central design consideration for both the usability and aesthetic appeal of the workspace is to use a visual linkage between the different shown components. For example, we always place the two topic models on the two sides of the screen, referring to them as the *left* and *right* model. In addition, we use a consistent color-reference every model (**orange** for the **left** model and **purple** for the **right** one). The color is also used to indicate similarity, e.g., **blue** is used to refer to **common keywords** and **green** is used to refer to a **document overlap**. A more subtle linkage is achieved by representing all topics consistently as dots and all documents as bars.



In addition to linking the visual encoding, we designed the visualization dashboard with stable visual anchors for non-changing components between views. We rely on sweeping animated transitions between the different views and keep non-changed components anchored to preserve them as reference points for the users' analysis. Users are guided by the consistent layered interaction model, where they *peel off* layers to go deeper into the analysis. At any time users can switch to higher-overview layers and go back to pick-up their analysis where they left off. This is facilitated by attribute sorting, selections, and filtering which are globally effective across all levels of the view.

This visual workspace is tailored to the four tasks introduced in Sect. 1: getting an overview of the topic modeling output, understanding the topic descriptors, examining the corpus feature distribution, and adapting the topic models through document relevance feedback. In this section, we discuss the design of the four views of the visual workspace, each corresponding to one task.

4.1 Topic Matching

As discussed by Alexander and Gleicher [2], comparing the results of two topic modeling algorithms through aligning their results is one of the most important tasks to get an overview of the results of topic modeling algorithms. Therefore, the entry point to our visual analysis workspace is the topic matching view, as shown in Fig. 3. This visualization relies on the output of an automatic topic matching approach [19] that identifies three levels of topic matches: complete

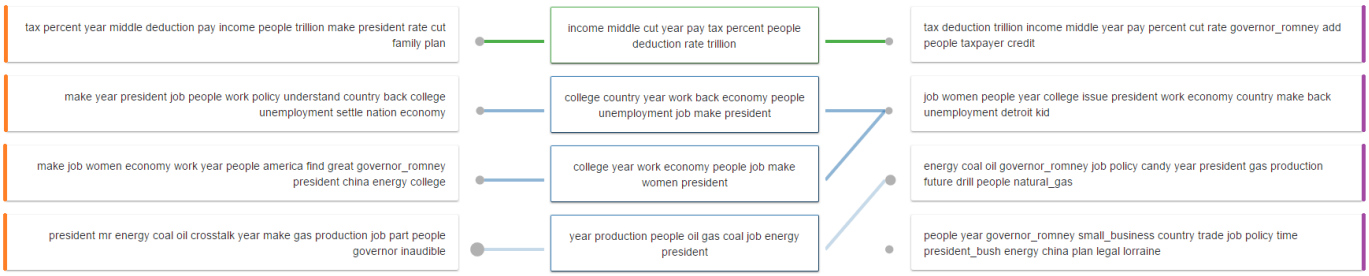


Fig. 3. *Topic Matching View.* Two different LDA topic models of the presidential debate between Obama and Romney in 2012 are shown on the left and right side. Their matches, ordered by decreasing match similarity (min. match similarity 0.7) while minimizing edge crossings, are shown in the middle. Green lines indicate a **complete match**, blue lines a **similarity-only match**. The gray points encode the number of documents in a topic. Multiple edges coming to one point indicate, that the respective model combined multiple topics of the other model into one.

matches, similarity-only matches, and mismatches. These levels are computed based on two criteria: the topic descriptor similarity and the underlying document overlap. Hereby the descriptor similarity between two topic descriptor vectors is computed using the **R**anked and **W**eighted **P**enalty **D**istance function (RWPD) that introduces a weighted penalty for every keyword that is present in only one of the two descriptor vectors. With an initial distance of 0, the final distance between the descriptor vectors of two topics ($topic_1$ and $topic_2$) is computed as follows:

$$\forall i \in topic_1, \forall j \in topic_2 : \begin{cases} i = j : \frac{|i-j|}{n} \times r \times (w(n,i) + w(n,j)) \\ i \notin topic_2 : w(n,i) \times p \\ j \notin topic_1 : w(n,j) \times p \end{cases}$$

$$w(n, pos) = \{ n > pos : \sqrt{n} - \sqrt{pos} \quad ; \qquad else : \sqrt{n} - \sqrt{n-1}$$

with p representing the added penalty; n as the minimum vector length of $topic_1$ and $topic_2$; r as the maximum distance range; and i, j as descriptor keywords from $topic_1$ or $topic_2$, respectively. This distance function was developed to mimic the human perception of ranked descriptor vector similarities [19].

A complete match (represented in green) between two topics fulfills both criteria of the algorithm: it has a high descriptor similarity and a significant document overlap, whereas the document overlap in similarity-only matches (depicted in blue) is not substantial. Mismatches (shown in yellow) are defined by a high document overlap accompanied by a low descriptor similarity. In this view, the similarity threshold for topic matches can be varied interactively by the user, depending on the level of granularity of the analysis. These different levels of matchings are used by analysts to identify the similarities between topics on the two considered levels. For example, two topics might share a large number of keywords but no document overlap, revealing a disagreement in the document-topic assignment between the two compared models. In addition to the level of matching, this view also highlights relationships between the topics in both models, such as splitting, merging, matching, and absent topics.

Fig. 3 shows an example of the topic matching view. In this view, the left and right topic models are aligned based on their matching score. Every topic from the two models is represented as a box containing the ranked descriptor vector of that topic. On the inner side of the box is a circle that is scaled to the number of documents assigned to that topic. In the center of the view, the topic matches are shown with their respective color, indicating the matching level (green, blue, yellow) and the matching score which is mapped to opacity (with lower values being more opaque). The topic matches show the ranked set of common topic descriptors. To minimize edge crossings, the position of every topic is determined based on a priority queue that favors larger topics with higher match scores followed by topics that match an already displayed one. By hovering over a topic match, the common keywords of the two matching topics are highlighted in bold and the matching score is shown as a tool-tip. In order to inspect a single topic or a pair of matching topics further, the user navigates to the next view by clicking on the object of interest.

4.2 Topic Summarization

The second view of our workspace is the topic summarization view depicted in Fig. 4. The main purpose of this view is to generate a better understanding of the topic descriptors in order to facilitate the interpretation of a topic. This is done through displaying the most significant sentences from the documents assigned to each topic as its summary. The number of shown sentences is set to ten by default but can be adjusted by the user. These sentences are chosen to assemble a representative summary of a given topic using a tailored scoring routine. The score for topic t_i and sentence s_j is calculated as follows:

$$score(t_i, s_j) = \frac{\sum_{x \in \{w \mid w \in s_j\}} sig(x)}{\max(\bigcup_{s \in t_i} \sum_{y \in \{w \mid w \in s\}} sig(y))}$$

In other words, for every sentence, the score sums up the significance values of all unique keywords and normalizes them to the highest score of the most representative sentence of the particular topic. Using only unique keywords counteracts potential skewness towards long sentences or repetitive phrases. The keywords considered are all descriptors of the topic at hand and their significance value is given through the topic modeling algorithms and the document descriptor scoring. This scoring function determines a ranking among all sentences attributed to documents that belong to a certain topic. However, in order to assemble a representative collection of sentences to summarize a topic, we strive to maximize their diversity. This is achieved by introducing a penalty function for the selection of the representative sentences to display in the visualization. Given the number of sentences to be shown as summaries, the function penalizes sentences which consist of exactly the same keywords as previously extracted sentences, i.e., for every set of similar sentences only the one with the highest score is displayed. This guarantees the needed diversity within the topic summaries in order for them to be representative and maximizes the number of topic descriptors shown across the selected sentences.

Fig. 4 depicts the design of the topic summarization view. In order to remain consistent with the visual encoding of the workspace, the two topic models are assigned to the left or right side, respectively. Hence, this view shows a mirrored visualization for the two models. Since this view is a direct transition from the topic matching view, the topics are represented using the same circles (now colored) from the previous view. To ensure linking these circles to the topics shown in the previous view, the topic summarization view is opened using an animated transition that moves the results of the two models to their corresponding sides in a sweeping motion, hiding the labels and keeping the dots. The mirrored visualization of the topic summarization view consists of a central bar chart representing all documents of the corpus, two title panels for displaying descriptors of the topic currently in focus, as well as two central panels for showing the topic summaries. In order to display a topic summary, the corresponding topic has to be pinned (◎, ◎) by selecting its representative circle. Pinning a topic has the effect of keeping them fixed and not updating the elements of the visualization through hovering. When pinned, the descriptions and summaries of the given topics are loaded in their corresponding

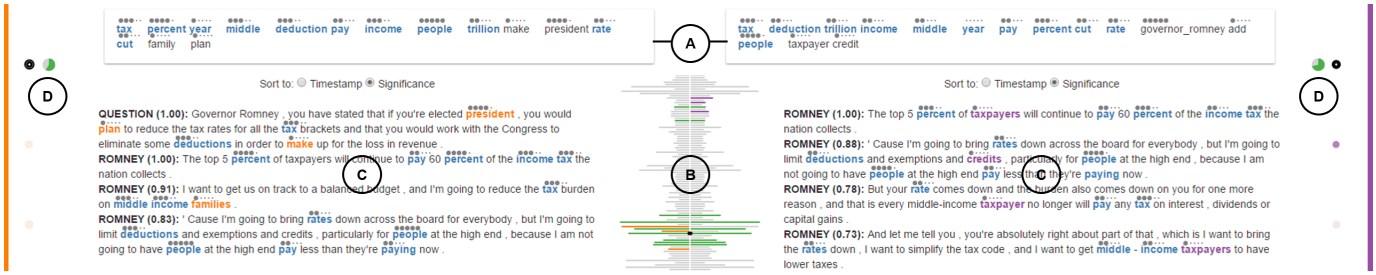


Fig. 4. *Topic Summarization View.* The topic descriptors of the two compared topics are shown in the top cards on the left and right Ⓐ. Descriptors appearing in both topics are colored blue, those appearing in only one topic orange or purple, respectively. All keywords are associated a small glyph above them, showing their relevance score for the topic. The mirrored bar chart in the middle Ⓑ shows all documents in the corpus, where the length of the bars is mapped to the length of the document. Documents belonging to the respective topic of both models are colored green, documents appearing in only one of both are orange or purple. To the left and the right of the bar chart, the top 10 most representative sentences for the topics are shown Ⓒ. The pie charts Ⓓ show the percentage of matching documents of the topics.

panels. In addition, the mirrored-bar charts highlight the documents that are assigned to the pinned topic. Upon pinning two topics on either side, the corresponding document overlap is shown in green and their matching descriptors are highlighted in blue. Additionally, small pie charts indicate the model agreement: the fraction of documents shared between topics in relation to the amount of documents assigned to the pinned topics, e.g., vs. for the left and right topics of Fig. 4, respectively. If the topic pinning is toggled off, the visualization is continuously updated when hovering over a topic circle or a document bar in the central bar chart.This feature becomes useful when exploring all relations to a given entity.

The central bar charts are an essential component of the whole workspace. Apart from the topic matching view, these charts are used in some form in every visualization to navigate through all the documents of the corpus. Through hovering over a document bar, a position indicator (small black dot) is updated to the document's position and this document is shown in the close-reading view on the bottom of the workspace (not captured in the screenshots). By default, all documents are ordered sequentially according to their order in the corpus. However, to enhance the exploration and understanding of the topic-document relationships, the document bar charts can be reordered globally with respect to a selected measure or the overall length of the documents. For a deeper understanding of the measure and parameter distributions, the user can switch to the next view which is designed for the exploration and analysis of parameter distributions.

4.3 Parameter Distribution Analysis

Fig. 1 shows an excerpt of the parameter distribution view which uses comparative bar charts to enable the efficient comparison of multiple feature and parameter distributions across the corpus. As mentioned in the previous section, the mirrored bar charts allow the navigation through the corpus while highlighting the document-topic assignments. Hence, after understanding the topic compositions, the user can further dive in the investigation of the documents by switching over to this view. An animated transition splits the topic summarization view along the central mirrored bar chart, moving the components to the left and right edge of the screen to peel off another layer. This parameter distribution view goes deeper into the structure of the corpus and allows the exploration of patterns across all document features and parameters. In keeping up this metaphor of a layered analysis, our visual analytics workspace allows the user to go up to any overview visualization at any time, then switch back to continue where they left off at a deeper layer.

To enhance the comparability of features across the corpus, we designed the comparative bar charts. This visualization technique is simple, yet has proven to be effective for the comparative analysis of two ordered distributions with an underlying baseline. As shown in the side-figure, we display the baseline distribution using mirrored dark bars in the background to be a constant reference for comparison. On top, the ratio of the two compared distributions dictates the interpolation range of the violin plots that are spanned asymmetrically on either side of the symmetry axis. Hence, for two given parameter values for each of the topic models, we calculate their proportion and use the larger ratio as maximum for the normalization of the opaque violin plots. Consequentially, the smaller value will not cover the baseline bars, leaving the size of the peaking-out bars as indicator for the relative difference between the two parameter values.

Using this visualization technique, we can arrange all relevant parameters and features for comparison, as shown in the side-figure. The document bar charts corresponding to the two topic models are situated on either side of the parameter distribution plots. Similar to the previous view, the document bar charts are used to navigate through the corpus. By hovering over a document bar, the close-reading view (not shown in the figure) is updated, as well as the navigation line which shows the concrete values of the baseline distribution for the particular document. These values could be chosen to represent the absolute frequencies of the features for every document. However, by default, they show the number of occurrences of a feature divided by the total number of words per document. This default value is chosen to emphasize the importance of that feature for the classification of the document. For example, if a ten-word document contains five adjectives, then varying the weight of adjectives will significantly impact this particular document in contrast to a longer document with the same amount of adjectives. In addition to linking and brushing, this visualization supports pinning topics and sorting all documents according to any feature or parameter. This becomes especially useful when choosing which documents to inspect further in the next view.

4.4 Document Relevance Feedback

This view is the main interface for decision making in order to adapt the topic modeling results in the next cycle of the progressive learning process, as described in Sect. 5. While the previous three visualizations were focused on understanding and comparing the two topic modeling results, the document relevance feedback view requires the users to actively vote for the most suitable model using their acquired knowledge. This is done using the interface shown in Fig. 5. The document relevance feedback view is always present in a minimized form at the bottom of the workspace, serving as an interactive close-reading view.

To activate the functionalities for the relevance feedback, the minimized close-reading panel is transitioned from the bottom of the workspace to the center, preserving the visual linkage of the anchored topics. When extended, this view is accompanied by a *decision-slider* between the two anchored topic-label boxes, along with a horizontal document bar chart which is sorted and colored according to the selected global sorting measure. In the center of this visualization is the currently selected document, with its keywords highlighted. In order to start the relevance feedback, the user selects a document from the horizontal bar chart (or through navigating to the previous/next document using the arrow buttons) and moves the decision-slider to the topic that yields a better description of the selected document. Our framework is designed to accommodate individual optimization strategies that depend on the user's analysis goal, time-budget, expertise and

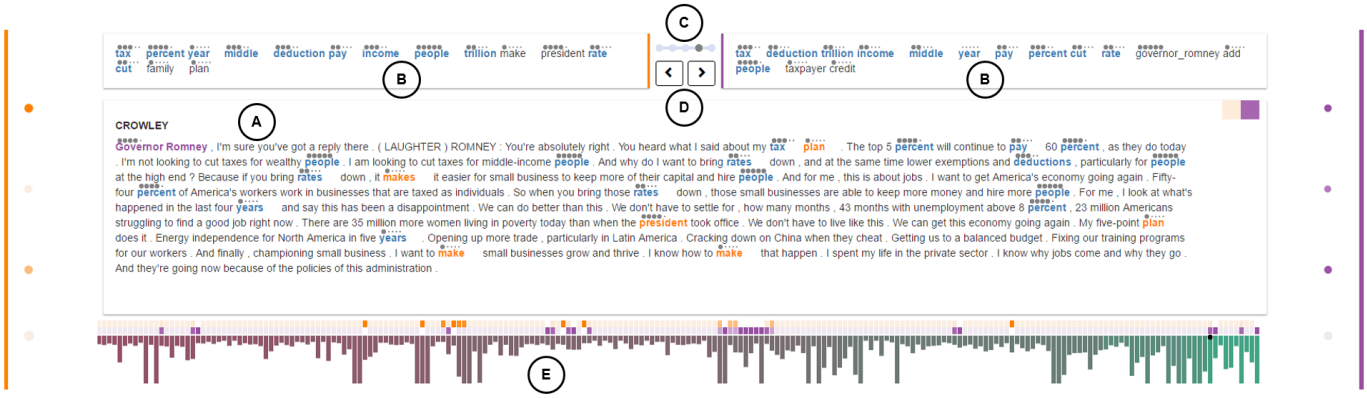


Fig. 5. The *Document Relevance Feedback View*. Ⓐ The document in review; Ⓑ the topic descriptors of the associated topics; Ⓒ the decision slider; Ⓓ the navigation arrows. The bar chart Ⓔ shows the documents sorted and color coded by their topic coherence, from bad (red) to good (green). Users rate topics for the current document by selecting a slider position, and can navigate between documents with the arrow buttons Ⓓ.

familiarity with the analyzed corpus, and the noisiness of the document collection. However, in order to assist the users in choosing representative documents for the relevance feedback task, we developed two quality metrics.

Document Quality Metrics In order to minimize the manual effort of the users, we introduce two quality metrics with the intention to direct users to the documents which will be most affected by their decisions. Given a document and its top topics from each model, these metrics assess the controversy among the two topics and between the document and the topics, respectively. The first metric is the topic coherence, which describes the agreement between two topics assigned to a document. The second metric is the topic certainty, which measures the compatibility of the two topics with the given document. We developed both metrics to comply with the human perception of similarity between two descriptor vectors and thus utilize the ranked and weighted penalty distance function (RWPD, introduced in Sect. 4.1) for the calculation of the measurements. Hence the two quality metrics are defined for every document as follows: $TopicCoherence = RWPD\ (topic_1, topic_2)$; $TopicCertainty = avg\ (RWPD\ (topic_1, doc),\ RWPD\ (topic_2, doc))$.

Decision-Making The decision-making process is guided by two basic questions that the user has to keep in mind. The central question for the document relevance feedback is *which topic describes the current document better?* Beyond the scope of a single document, the user can further consider the question: *Would you like to endorse or eliminate single descriptors to guide the topic modeling?* These questions are addressed by two interactions: the decision-slider and the single-word boosts. These interactions are directly translated to actions for the adaption and learning of a new model in the next cycle and are accompanied by an immediate visual confirmation to manifest the changes caused by the users' decision.

The decision-slider simplifies the model-steering to a binary decision between two models for every document. To accommodate the probabilistic nature of some topic modeling algorithms, this binary decision is extended to be based on the assigned topic probabilities. Regardless of the reviewing scheme, the user is only required to make singular decisions for one document at a time. The reinforcement learning algorithm then generalizes from the users' verdict on a sample of documents to the overall corpus. Therefore, it is essential to point the users to representative documents to consider for review. Our proposed optimization strategy is to examine outliers and documents with a high level of controversy, i.e., documents at the lower-end of the quality metric scale introduced in the previous section. Accordingly, a time-efficient and successful optimization strategy we observed, is to order all documents based on their certainty or coherence scores and examine longer documents with a low score for these measures.

When analyzing a document, a decision is made through moving the slider towards the better-suited topic on a discreet five-point scale. If both topic models have an equal quality regarding the analyzed document, the slider can be kept in the middle, in order to remain neutral. Moving the slider towards a topic implies a keyword boost in favor of that topic. As described in Sect. 5.1 in more detail, the boosting differentiates between three keyword types: matching keywords from both topics; keywords contained only in the favored topic; and keywords contained only in the rejected topic. The matching keywords are positively boosted (by a factor of two) since they are clearly agreed upon by the two models. The undesirable keywords from the rejected topic get a negative boost (by a factor of one), while the favored keywords from the endorsed topic receive the highest positive boost (by a factor of three). In addition to the keyword types, the weighting of the slider amplifies the keyword boosting in the case of a polarized decision.

Using the decision-slider affects the topic descriptors defined by the two models and leverages the users' preferred keyword compositions to sustain them. However, in order to accelerate the topic convergence, users might want to globally sanction or endorse single keywords. This is achieved through single-word boosts and penalties. Such a functionality gives the users a higher degree of freedom to optimize the topic modeling. However, through considering the significance of keywords when applying the boosts, our system counteracts potential overfitting of the descriptors. In contrast to the decision-slider, single-word boosts and penalties can be applied to any word (or n-gram) in the corpus. Therefore, beside spurring the topic convergence, this functionality is also extensively used to remove undesirable and nonsensical words from the topic descriptors in the further optimization cycles.

Visual Confirmation Since the interactions in the decision-making phase have a wide-ranging implication on the quality of the relearned model of the next processing cycle, we incorporated an immediate, responsive visual feedback for the actions performed in that step. On the one hand, we track the movement of the decision-slider for every document and represent its assigned topic preference using a small icon on the top right corner of the text panel, as well as in the horizontal document bar chart on the bottom, as depicted in Fig. 5. These icons use the two topic colors to highlight the favored model for every document, in addition to showing the certainty of the user's decision using opacity, e.g., . Additionally, we highlight documents that have been visited by the user (without scoring). On the other hand, we show the scoring of every keyword through simple glyphs on top of them, e.g., Supreme Court, Court, and Constitution. The glyph consists of five dots that represent the binned score of the keywords, e.g., ●●··· for a score of 2/5. This glyph is immediately updated when changing the slider to indicate the effect of the change on the keyword scoring. The score is determined by the keyword significance for the current topic or document, combined with the overall score of that keyword (which gets affected by the boosts). Having instant visual confirmations had a positive impact on the usability and understandability of the visualization when presented to users.

5 Task-Driven Topic Convergence

As shown in Fig. 2, the users' input from the Visual Analytics Workspace is used to update the inputs before starting a new topic modeling cycle. By endorsing topic descriptors or single keywords, the users are able to steer the topic modeling in order to converge towards a more intuitive and understandable topic modeling result. As described in more detail in the following sections, this goal is achieved through reinforcement learning in iterative optimization cycles.

5.1 Reinforcement Learning

Using the relevance feedback described in Sect. 4.4 we constantly learn and update feature weights throughout each cycle. Whenever the users rate the topic models with respect to a given document by moving the decision-slider, we update our data structures in the background in order to prepare for the next cycle. The resulting changes are immediately presented to the users and enable them to quickly understand the impact of their actions on the topic models, despite these changes only affecting the new topic modeling cycle. Before the first cycle begins, each topic keyword in the corpus is assigned an initial score. While this value is equal for all keywords in the current implementation, it can be adjusted by the users. However, it is important that this score be larger than zero, as the scores of the document descriptors are adjusted proportionally to this value before the topic modeling algorithms are instantiated.

Each update of the decision-slider is reflected in the scores of the affected keywords: topic descriptors of the rejected topic model are penalized, while topic descriptors of the favored model are boosted by three times the value of the penalty. Keywords appearing in both sets of topic descriptors are boosted and penalized at the same time, resulting in a boost by two times the penalty. These ratios for boosts and penalties have yielded promising results in our experiments, but could be easily adjusted by the users to quicken or repress the learning rate. The decision-slider has five possible positions to allow users to show a strong or subtle preference of one topic model over the other. The left- and rightmost positions of the slider correspond to a strong preference of the left or right model, respectively, and lead to boosts and penalties twice as high as the values for the more subtle preferences mentioned above. In the middle position, the scores are not updated.

While boosting and penalizing existing topic descriptors is already a very powerful tool, it is not always sufficient. An additional feature that has been often used during our user studies is the option to promote arbitrary words from any document, that have not yet been recognized as a good topic descriptor by the current models. In the same way, (key)words can be penalized to ensure that they will not be part of the set of topic descriptors in a future run. These *single-word boosts* give the users a very direct way of incorporating their domain knowledge and correcting inherent biases of the topic models which would otherwise be very hard to compensate for. When promoting single words, we first make sure to boost them to the base score associated to topic keywords, before adding another boost of three times the maximum boost that can be achieved by one slider movement.

Between the runs of two cycles we use reinforcement learning to update all parameters, as described in more detail in Sect. 5.2. While in the current implementation the learning rate is fixed to a constant 40%, our framework can easily accommodate more complex and sophisticated learning strategies. For example, the "Win or Learn Fast" principle has previously been successfully applied in user driven topic modelings by Tripolitakis et al. [40]. In case of the results being positively rated by their users they reduce the learning rate, and drastically increase it whenever performance worsens. Additional random changes in the parameters occurring with very low probability help to escape local maxima. Thanks to the modular nature of our framework such an extension could easily be included in the future.

As part of our future work, we also plan to add boosting of semantically similar words as determined by word2vec [29]. Instead of only updating the scores of topic descriptors as a result of a decision-slider movement, we plan to calculate their most similar words and to boost them by half the boost value of the respective similar topic descriptor. Of course, the exact weight proportions can easily be adjusted by users. This will help to avoid overfitting the topic model for single documents that have been rated by the users, and will instead lead to a more general shift in topic assignments for all documents in the corpus. As a consequence, the workload for the users could be reduced, as less documents have to be rated in order to achieve a good training result.

5.2 Iterative Topic Evolution

To ensure the robustness of our progressive learning process and to guide the users through the optimization space, our technique is designed to relearn only the under-performing topic model while keeping the better model as an anchor for the next cycle. When the users decide to finish a cycle and restart the processing loop, we assess all slider positions chosen by the users in order to determine which topic model has—according to the user—performed better on the given corpus. For example, assuming the overall average slider position was on the left, meaning the users preferred the left topic model, we keep this model as a baseline for the next cycle and recompute the right one. Before restarting the topic modeling, we adjust the input by applying the boosts computed in Sect. 5.1 to all words of all documents. This is done using the document descriptor generator component (Sect. 3.2) by directly adjusting the computed frequencies of every word, or repeating it in the input text before re-applying the scoring functions, which has the equivalent effect. Words that have been penalized a lot and, as a result, are associated with a zero or even negative score, will be removed by the document descriptor generator and are not taken into account for the topic models of the new cycle.

In addition to boosting keywords, we also update the feature weights (Sect. 3.1) for the under-performing model, i.e., the influence of word classes on the topic model. After each update of the decision-slider through the users, we retrieve the associated features of the topic descriptors in order to update the ratio of boosted features. This is done independently for the two active topic models. Before we start a new cycle, we collect the ratios of feature weights of the better model, and use them to update the feature weights of the under-performing model. This update happens with a user-adjustable learning-quota that is initially set to 40% to ensure fast convergence after a limited number of cycles. Between two cycles, all collected data, as well as key metrics, such as topic coherence and certainty scores, topic assignments on a per-document basis, and the current corpus keyword scores, are persisted to a log file. Such detailed data collection enables the evaluation of the topic model at a later stage and makes provenance tracking possible. Additionally, this data is of further interest for users, as expressed by a political scientist during our user study, who would like to further analyze the details of the topic development between cycles.

5.3 Global Learning of Parameter Scales

One disadvantage of the "educated guess" for feature weights as introduced in Sect. 3 is the fact that they are initially extracted on a per-corpus base. As a result, they are relatively susceptible to changes in linguistic nuances, such as writing styles, that are specific to a given corpus. We counteract this bias towards certain features by introducing a set of globally trained feature weights. These global weights are automatically updated after a successful set of training cycles has been finished by the users. In the current implementation the global feature weights are updated using a relatively low learning rate. However, different, more complex strategies such as emphasizing more recent runs could easily be added thanks to the modularity of our framework.

Before starting the first cycle the users are presented with the feature weights that have been automatically extracted for their given corpus. They can then either use them as is, or leverage their domain knowledge to decide that they are a non-optimal fit. In this case they can either manually adjust the weights, or mix in the global feature weights by a definable percentage. This leads the descriptor extractor to emphasize certain features and consequently to a topic model of higher quality and understandability. Additionally, it enables users to start the feedback cycle without having to pre-process their corpus, by only using the global feature weights.

Once the users see the current better-performing topic model as a good fit for their use case, they might decide to finish the feedback loop. They are then given the option to update the previously learned global

feature weights with the ones they just trained for the better-performing model. In case they decide to do so, the global weights are adjusted for the newly learned weights with a user-definable learning-quota.

As part of our future work this concept could be extended in order to maintain multiple "global" feature weights for different classes of documents. In this case, the users would have to assign one or multiple classes to their current corpus, before updating the respective weights. This would be particularly interesting for users working on different document classes, such as political speeches, news stories, and books. We also plan to globally learn the scores associated to keywords described in Sect. 5.1 instead of restarting that process for every corpus. Guided by updates with a user-definable learning quota we plan to move away from uniform starting scores for all keywords, instead utilizing the results from previous runs as an improved starting point. Such a feature would be helpful for users analyzing similar corpora, e.g., multiple presidential debates, where otherwise the first runs will always be spent on retraining the same keyword scores.

6 Evaluation

Due to the modularity of our technique and the subjectivity of the interpretation of topic models, we chose to evaluate our framework with a mixed-methods study, as advised by Isenberg et al. [22]. We empirically measured the effectiveness of the progressive-leaning process with automatic metrics, as well as a manual assessment of model quality by expert annotators. We also gathered qualitative feedback on the usability of our visual analysis workspace.

Dataset To choose an appropriate dataset, we envisioned a corpus that fulfills three criteria. First, we wanted broadly-familiar content to ensure understanding by participants and annotators. Second, we sought document collections with shorter documents in order to fit multiple optimization cycles into a two-hour session (as the reinforcement learning scales in time with the corpus length). Third, in order to empirically validate the results of the study, we wanted a corpus with a known topic distribution as gold standard. Thus we chose to use a presidential debate for the study, specifically the second US presidential debate between Romney and Obama in 2012. This debate discussed known domestic affairs, has been widely studied and the topics are accessible to a non-expert reader. And lastly, it fulfills the final criterion if we consider the document granularity on an utterance level.

Controls In order to control our study to focus on model improvement through the iterative learning process, we controlled for the topic modeling algorithm and the initial parameter settings and model across all participants. We chose to run the study based on LDA, as it is the most common baseline across the literature. We initialized both topic models to LDA with 9 topics (determined during pilot testing), as using the same model on both sides leads to more predictable behavior. We initialized one with a feature weighting based on entropy and the other with a manually selected feature weighting expected to be helpful for this dataset (to simulate manual tuning of features, which is the normal process without our technique)[1].

Method and Participants Before conducting our formal study, we ran a pilot study with three graduate students to test all conditions and refine the usability based on their feedback. Our study was broken into two tasks, each completed by different participants. In the first stage, the *Model Improvement Task*, we conducted six two-hour-long sessions with 2 experts each from political science, linguistics, and computer science. Participants in this phase had varied experience with topic modeling from novice (computer scientists who had embedded topic models in tools, but not tuned them) to expert (political scientists who use topic models and manually tune them). In the absence of a standard benchmark, we created a second stage, the *Model Assessment Task*, in which we evaluated the outputs of the first stage using both automatic quantitative metrics, as well as manual quality coding carried out by three annotators from linguistics who were all knowledgeable about topic modeling and trained to recognize word relations.

[1] verbs, adj, adv, and $\overline{\text{stopwords}} = 0$ – everything else = 100

Model Improvement Task – This task centered around participants using the technique to perform an iterative optimization of the given dataset and models. Each study session was divided into three parts. In the first 30 minutes, we started by asking the participants about their experiences with topic modeling, then continued explaining the learning process and the visual interface design. This was followed by a brief initial feedback round to gather first impressions. In the second part (1 hour), we asked participants to use the visual interface in order to optimize the topic modeling results of the presidential debate in an iterative cycle. They were free to use all features of the interface and execute as many refinement cycles as they wanted. During this phase we asked participants to 'think aloud' as they worked with the interface, describing their choices and any usability problems they encountered. We also collected interaction logs, screen capture videos, and the topic models generated for each optimization round. In the final part of the study, we collected their feedback on the performance of our framework and their satisfaction with the results, considering the time spent on optimization.

Model Assessment Task – In this phase we invited annotators to assess the quality of the topic models generated in the first phase. After a brief introduction to the study annotators were given 12 worksheets, each listing 9 sets of topic keywords in rank order. Each worksheet represented the output of either an initial or final topic model from phase one (1 initial, 1 final, for each of the 6 phase one participants). The worksheets were not labeled and were presented in random order. Annotators were also given a selection of *gold standard* topics widely accepted to be represented in this dataset [42]. The annotation task was to examine each topic and determine the best matching gold standard topic, and give a rating from 0–4 of the topic match to the standard. Next, the annotator circled all words in the topic descriptor vector which did not fit the gold standard. This process was repeated for all topics and topic models (9 topics X 12 topic models X 3 annotators = 324 ratings). Due to the demanding nature of the task, we did not annotate intermediate topic model states. The measures collected in this phase were topic model precision change, average irrelevant words change, and interannotator agreement.

6.1 Quantitative Results: Model Improvement

We calculated the change in model certainty using the automatic certainty score described in Sect. 4.4. The results, charted across all iterations of reinforcement learning, are shown in Fig. 6. The average uncertainty (black line) decreased monotonically through each iteration. The uncertainty improvement achieved by participants varied, but for all of them the overall model uncertainty was lower at the end of the study than the beginning (average 35.3%). That is, the models consistently improved through the learning iterations. While there are too few participants to confirm a trend, we did observe a better improvement from participants who had more understanding in topic modeling before the experiment (linguists).

Expert scoring of topic precision also showed an improvement across all phase-one participants and all topics — the average before optimization was 2.31 ($\sigma = 0.46$) and after the last optimization cycle 3.68 ($\sigma = 0.67$) on a [0,4] scale, with an inter-annotator agreement on topic precision of 88.3%. For comparison across automatic and manual measures, Fig. 6 contains both metrics in a percent scale and reveals a strong agreement between them. The precision improvement by topic is also shown.The number of irrelevant words per topic decreased by 7.2% on average between the initial and final topic models, further indicating model improvement.

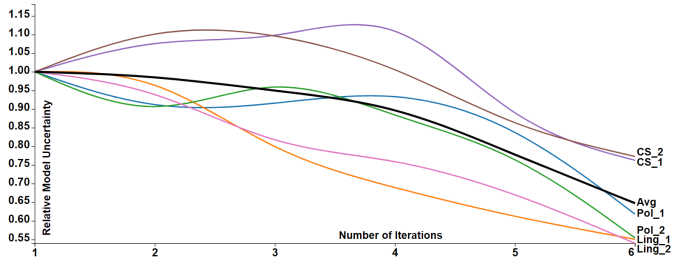
Due to the probabilistic nature of the topic modeling algorithm used in the study and with the lack of an annotated gold-standard dataset on the granularity of the examined documents, we can only make a reliable conclusion using a relative scale to achieve a stable baseline for measuring the effect of topic improvement over time. Hence, given these starting conditions, the decrease in the model uncertainty is statistically significant, however, due to the unavoidable variance in our baseline, we do not have the experimental power to make a statistically-reliable claim. Therefore, we present our quantitative results as relative changes over time indicating a trend of substantial model improvement over all

Participant	Topic Precision Improvement	Automatic Model Certainty Improvement
Pol_1	33.3%	38.2%
Pol_2	25.0%	44.6%
CS_1	38.3%	23.8%
CS_2	33.3%	22.7%
$Ling_1$	45.0%	45.0%
$Ling_2$	48.3%	46.0%
Avg	**37.2**%	**35.3**%

(a) Average improvements between the first and last optimization cycles for every participant.

Topic	Topic Precision Improvement
Taxes	33.3%
Unemployment	5.6%
Education	13.9%
Gun Law	19.4%
Energy	58.3%
Women's Rights	25.0%
Immigration	36.1%
Attack in Libya	72.2%
Crosstalk	8.3%
No Topic	-100.0%
Avg	**37.2**%

(b) Average improvements between the first and last optimization cycles by topic.



(c) The *Automatic Model Uncertainty* shows the positive development of the topic assignments, especially when trained by experts. Values at itr. 6: $CS_1 = 0.76$; $CS_2 = 0.77$; $Pol_1 = 0.61$; $Pol_2 = 0.55$; $Ling_1 = 0.55$; $Ling_2 = 0.53$; $Avg = 0.64$.

Fig. 6. User study results indicating a clear improvement for all participants and topics. These are based on an empirical analysis of the logged data.

optimization cycles. This trend was confirmed in our qualitative results.

6.2 Qualitative Results: Expert Feedback

Initial Feedback Regardless of their expertise in using topic models, all participants of the study saw an immediate benefit in having such a visual analytics process. Political scientists, who had more experience with topic models, reported that they often spend hours in a trial-and-error cycle to get satisfying results. When asked about their usage of topic models, computer scientists reported that they heavily relied on the automatic output of topic models to embed topics in their tools, not considering model uncertainties or fitness to the data. However, all participants uniformly agreed that one of their major concerns with topic models is the reliability of the outcome and trustworthiness of the black-box. Especially the linguists were mistrustful. One of them commented that she is unaware of successful optimization strategies that would help her validate topic modeling results using her data.

Visualization Design and Usability Participants appreciated the visual anchoring we employ throughout the different stages as it gave fix-points to concentrate on and helped in guiding them through the process. Especially in combination with the layered analysis allowing them to work on a higher level, or get more detailed information on demand, it enhanced the orientation during the visual analysis. They especially liked the steadily-visible, interactive close-reading panel on the bottom of the view for keeping the analysis in context. One feature that was extensively used by all participants of the study was the option to boost or penalize individual words as a form of direct relevance feedback to influence the topic model results towards being more intuitive. This confirms the preferred feedback mechanisms discovered by Lee et al. [25]. It was especially useful to boost words that had not yet been recognized as good topic descriptors by any of the models.

Most users noted the steep learning curve due to the diverse functionality, the number of included visualizations and their rich set of interactions, and the density of the available information. However, all participants were able to achieve proficiency with the tool over the course of the study session. One political scientist (Pol_1) commented that "in order to have such an expressive visualization dashboard for the analysis [he is] willing to take into account *learning* to use a new system." He added, "if we establish such a framework as a norm for that analysis and use of topic models [he expects] that our visual workspace will be improved and extended by the demands that would arise from a community of active users".

Although appreciating the degree of freedom in the analysis and the serendipity of individual optimization strategies, one of the computer scientists (CS_1) suggested incorporating an option for more guided optimizations through adding explicit system recommendations. This trade-off between serendipitous and guided discovery for the analysis of topic models has been exploited by Alexander et al. [3], who argue for a more open and serendipitous analysis and exploration process.

General Assessment After using the tool for a while the users reported they were *unobtrusively* learning more about the dataset due to the effectiveness of the workspace design and the richness of on-demand information. User POL_1 positively commented that he "could spend hours exploring a dataset with that interface".

For the users it was intuitive to keep the better performing topic model, while restarting the one that had underperformed. It gave them an easy way to track the changes introduced in the new cycle. We noticed that most of the times users preferred the relearned model over the best performing one of the previous cycle, indicating that they immediately noticed an improvement in the topic model with every iteration. Additionally, all users predicted that they would expect even better results with some additional optimization cycles. They also noted that they deemed the trade-off between the additional time needed to complete a new cycle and the resulting benefit justifiable and reported that they would use the tool regularly on their own data.

7 Conclusion

In this paper we have presented a modular visual analytics framework for the progressive learning of topic modeling parameters. Our technique supports a layered analysis for the deep comprehension and adaption of topic models based on the data and tasks of the users. This layered analysis is provided through a Visual Analysis Workspace, consisting of four visualizations that are tailored to the analysis tasks of topic matching, topic summarization, parameter distribution analysis, and document relevance feedback. The workspace is backed-up by a reinforcement learning feedback loop enabling users to optimize topic models and obtain more easily understandable results. We have empirically verified that users at a variety of expertise levels can improve topic model quality using this human-in-the-loop process.

A web-tool implementing our visual analytics Workspace will be made available to the public for non-commercial use as part of the VisArgue project [20]. Through this, the optimized models can be used in other visualizations and systems. Additionally, the logged data for each refinement cycle will be available to download for further analysis and for the use in other computation models. Since the computationally-expensive steps of the analysis are performed on the server-side, our framework scales to the analysis of larger document collections. In addition, the modularity of the approach allows users to select a topic modeling algorithm suitable for the data and task at hand. Hence, for every given setting, different models can be used for optimization.

As future research, we would like to look deeper into the "black box" of topic modeling by going beyond parameter adjustments. As a form of a more direct model-steering, for algorithms that support it, we plan to let the users take a closer look at the topic modeling process while it is running and let them directly incorporate feedback that would immediately affect the remainder of the modeling process. We are also working on implementing an option to present users with guidance on refinement actions and optimization possibilities which would most likely result in model improvements, based on the internal model stress level and certainty. We are planning to make this guidance more tailored to the data and task at hand through active learning from the user interactions and relevance feedback.

REFERENCES

[1] C. C. Aggarwal and C. Zhai. A survey of text clustering algorithms. In *Mining Text Data*, pages 77–128. Springer, 2012.

[2] E. Alexander and M. Gleicher. Task-driven comparison of topic models. *IEEE Trans. on Visualization and Computer Graphics*, 22(1):320–329, 2016.

[3] E. Alexander, J. Kohlmann, R. Valenza, M. Witmore, and M. Gleicher. Serendip: Topic model-driven visual exploration of text corpora. In *Proc. IEEE Symp. on Visual Analytics Science and Technology (VAST)*, pages 173–182, 2014.

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

[5] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proc. Int. Conf. on Machine learning*, pages 113–120. ACM, 2006.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *J. Machine Learning Research*, 3:993–1022, 2003.

[7] J. Chang, S. Gerrish, C. Wang, J. L. Boyd-Graber, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 288–296, 2009.

[8] J. Choo, C. Lee, C. K. Reddy, and H. Park. UTOPIAN: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):1992–2001, Dec. 2013.

[9] J. Chuang, S. Gupta, C. D. Manning, and J. Heer. Topic model diagnostics: Assessing domain relevance via topical alignment. In *ICML (3)*, pages 612–620, 2013.

[10] J. Chuang, C. D. Manning, and J. Heer. Termite: Visualization techniques for assessing textual topic models. In *Proc. of Int. Conf. on Advanced Visual Interfaces*, pages 74–77. ACM, 2012.

[11] J. Chuang, D. Ramage, C. Manning, and J. Heer. Interpretation and trust: Designing model-driven visualizations for text analysis. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pages 443–452. ACM, 2012.

[12] C. Collins, F. B. Viegas, and M. Wattenberg. Parallel tag clouds to explore and analyze faceted text corpora. In *Proc. IEEE Symp. on Visual Analytics Science and Technology (VAST)*, pages 91–98. IEEE, 2009.

[13] W. Cui, S. Liu, L. Tan, C. Shi, Y. Song, Z. Gao, H. Qu, and X. Tong. Textflow: Towards better understanding of evolving topics in text. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):2412–2421, Dec. 2011.

[14] W. Cui, S. Liu, Z. Wu, and H. Wei. How hierarchical topics evolve in large text corpora. *IEEE Trans. on Visualization and Computer Graphics*, 20(12):2281–2290, 2014.

[15] M. Dash and H. Liu. Feature selection for clustering. In *Proc. Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, pages 110–121. Springer, 2000.

[16] M. Doerk, D. Gruen, C. Williamson, and S. Carpendale. A visual backchannel for large-scale events. *IEEE Trans. on Visualization and Computer Graphics*, 16(6):1129–1138, Nov. 2010.

[17] W. Dou, X. Wang, R. Chang, and W. Ribarsky. ParallelTopics: A probabilistic approach to exploring document collections. In *Proc. IEEE Conf. on Visual Analytics Science and Technology (VAST)*, pages 231–240, Oct. 2011.

[18] W. Dou, L. Yu, X. Wang, Z. Ma, and W. Ribarsky. HierarchicalTopics: Visually exploring large text collections using topic hierarchies. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):2002–2011, 2013.

[19] M. El-Assady. Incremental Hierarchical Topic Modeling for Multi-Party Conversation Analysis. Master's thesis, University of Konstanz, 2015.

[20] M. El-Assady, V. Gold, A. Hautli-Janisz, W. Jentner, M. Butt, K. Holzinger, and D. A. Keim. VisArgue : A visual text analytics framework for the study of deliberative communication. In *Proc. Int. Conf. on the Advances in Computational Analysis of Political Text*, pages 31–36, Zagreb, 2016.

[21] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. Themeriver: Visualizing thematic changes in large document collections. *IEEE Trans. on Visualization and Computer Graphics*, 8(1):9–20, 2002.

[22] T. Isenberg, P. Isenberg, J. Chen, M. Sedlmair, and T. Möller. A systematic review on the practice of evaluating visualization. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):2818–2827, Dec. 2013.

[23] S. Jänicke, G. Franzini, M. F. Cheema, and G. Scheuermann. On close and distant reading in digital humanities: A survey and future challenges. *The Eurographics Association*, pages 83–103, 2015.

[24] T. K. Landauer and S. T. Dutnais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, pages 211–240, 1997.

[25] T. Y. Lee, A. Smith, K. Seppi, N. Elmqvist, J. Boyd-Graber, and L. Findlater. The human touch: How non-expert users perceive, interpret, and fix topic models. *Int. J. Human-Computer Studies*, 105:28–42, 2017.

[26] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. Cambridge University Press, 2008.

[27] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999.

[28] A. K. McCallum. MALLET: A machine learning for language toolkit, 2002. http://mallet.cs.umass.edu.

[29] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 3111–3119, 2013.

[30] D. Mimno, W. Li, and A. McCallum. Mixtures of hierarchical topics with Pachinko allocation. In *Proc. of Int. Conf. on Machine Learning*, pages 633–640. ACM, 2007.

[31] D. Oelke, H. Strobelt, C. Rohrdantz, I. Gurevych, and O. Deussen. Comparative exploration of document collections: a visual analytics approach. In *Computer Graphics Forum*, volume 33, pages 201–210. Wiley Online Library, 2014.

[32] D. Ramage, E. Rosen, J. Chuang, C. D. Manning, and D. A. McFarland. Topic modeling for the social sciences. In *Proc. Advances in Neural Information Processing Systems (NIPS), Workshop on Applications for Topic Models: Text and Beyond*, 2009.

[33] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proc. Conf. on Uncertainty in Artificial Intelligence*, pages 487–494. AUAI Press, 2004.

[34] G. Salton, A. Wong, and C.-S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.

[35] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. Visual parameter space analysis: A conceptual framework. *IEEE Trans. on Visualization and Computer Graphics*, 20(12):2161–2170, 2014.

[36] A. Singhal. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.

[37] A. Smith, T. Hawes, and M. Myers. Hiérarchie: Interactive visualization for hierarchical topic models. In *Proc. Workshop on Interactive Language Learning, Visualization, and Interfaces*, page 71, 2014.

[38] K. Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[39] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *J. American Statistical Association*, 101(476):1566–1581, 2006.

[40] E. Tripolitakis and G. Chalkiadakis. Probabilistic topic modeling, reinforcement learning, and crowdsourcing for personalized recommendations. In *Proc. European Conf. on Multi-Agent Systems*, 2016.

[41] F. Wei, S. Liu, Y. Song, S. Pan, M. X. Zhou, W. Qian, L. Shi, L. Tan, and Q. Zhang. TIARA: A visual exploratory text analytic system. In *Proc. ACM Int. Conf. on Knowledge Discovery and Data Mining*, KDD '10, pages 153–162. ACM, 2010.

[42] Wikipedia. United States presidential debates, 2008 — Wikipedia, The Free Encyclopedia, 2017. [Online; accessed 31-March-2017].

[43] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proc. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 267–273, 2003.