This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2020.2981456, IEEE Transactions on Visualization and Computer Graphics

ArchiText: Interactive Hierarchical Topic Modeling

Hannah Kim, Barry Drake, Alex Endert, and Haesun Park

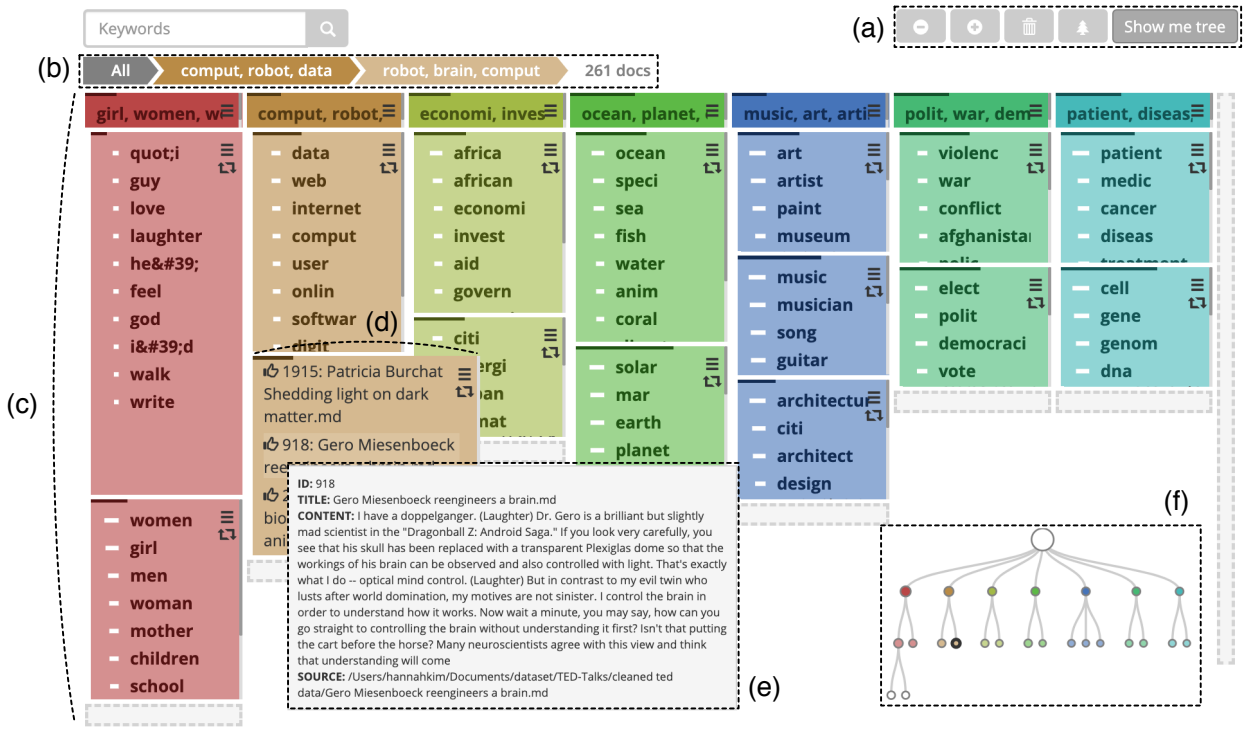


Fig. 1: The ArchiText system. The topic workspace mode has (a) a control bar, (b) a breadcrumb view, (c) a topic card view, and (f) a mini overview. The topic card view shows topic cards, which can be flipped to show (d) documents in the selected topic. (e) A detail view pops up when the mouse hovers over a document.

Abstract— Human-in-the-loop topic modeling allows users to explore and steer the process to produce better quality topics that align with their needs. When integrated into visual analytic systems, many existing automated topic modeling algorithms are given interactive parameters to allow users to tune or adjust them. However, this has limitations when the algorithms cannot be easily adapted to changes, and it is difficult to realize interactivity closely supported by underlying algorithms. Instead, we emphasize the concept of tight integration, which advocates for the need to co-develop interactive algorithms and interactive visual analytic systems in parallel to allow flexibility and scalability. In this paper, we describe design goals for efficiently and effectively executing the concept of tight integration among computation, visualization, and interaction for hierarchical topic modeling of text data. We propose computational base operations for interactive tasks to achieve the design goals. To instantiate our concept, we present ArchiText, a prototype system for interactive hierarchical topic modeling, which offers fast, flexible, and algorithmically valid analysis via tight integration. Utilizing interactive hierarchical topic modeling, our technique lets users generate, explore, and flexibly steer hierarchical topics to discover more informed topics and their document memberships.

Index Terms—Text analytics, topic modeling, nonnegative matrix factorization, hierarchical topics, visual analytics

1 Introduction

Analysis of large-scale text collections has been a widely studied research topic in the data analytics community. In particular, it is challenging to obtain an effective overview of text data and discover useful insights without going through each data item. This is untenable due to the sheer volume and the noisy, unstructured nature of text data. To solve this, various computational topic modeling techniques such as Probabilistic Latent Semantic Analysis (PLSA) [21], Latent Dirichlet Allocation (LDA) [2] and Nonnegative Matrix Factorization (NMF) [31] have been developed in recent decades. These methods provide content overviews by computing semantically meaningful topics as keyword distributions, and organize documents within the topics.

Recent advances in topic modeling have resulted in many new formulations and algorithms. However, even with the advances in topic modeling methods, results generated by these completely automated computational approaches depend largely on the problem formulation involving some objective functions and the corresponding algorithms. As a consequence, the computed results do not always match human expectations or context, and can be of poor quality, or difficult to make sense of [25, 33]. Human-in-the-loop approaches can be highly benefi-

- *Hannah Kim, Alex Endert, and Haesun Park are with Georgia Institute of Technology, USA. E-mail: {hannahkim, endert, hpark}@gatech.edu.*
- *Barry Drake is with Georgia Tech Research Institute, USA. E-mail: barry.drake@gtri.gatech.edu.*

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

cial for generating higher quality topics that align with users' needs and domain expertise. Furthermore, interactive exploration can be critical to foster understanding through discovery [42].

Many visual analytics systems for text data have been developed for human-in-the-loop topic modeling. These systems present a high-level overview of the text data by visualizing the topics generated by automated topic modeling algorithms and allow users to explore items of interest and, if possible, steer the underlying model. However, there are two major challenges in existing interactive topic modeling systems: *scalability* and *limited steerability*. First, most existing interactive topic modeling approaches can handle hundreds or thousands of documents, but they are not suited for large-scale datasets in real world scenarios. One reason is that in many systems, the underlying topic model is treated as a black box, which is recomputed from scratch after each interaction. In order to utilize human-in-the-loop topic modeling in real world applications, we need an efficient and scalable way to interact with a massive set of documents. Next, many visual analytics systems are often not well-suited for interactive analysis since they are built on top of existing automated topic modeling methods. In other words, interaction capabilities offered in these systems are limited by the convenience of the adopted methods [33]. An example is LDA [2], one of the most celebrated methods in topic modeling. Despite its popularity, LDA has several issues which hinder its integration with visual analytics systems. For instance, its parameters are not easy to understand and tune for non-experts. Sometimes even a small parameter change results in unpredictable side effects [33]. Also, LDA results are less consistent over different runs [8], which makes it difficult for users to trust the model and to see if interactions are properly reflected.

In this paper, we propose *hierarchical* topic modeling in contrast to *flat* (non-hierarchical) topic modeling in the context of visual analytics. Flat topic modeling is limited for visualizing large-scale text data. As the text data and vocabulary grow larger, the need for interacting and visualizing a larger number of documents and topics also grows, and it becomes more challenging to better represent the underlying data. However, since the computation of a very large number of topics at once is limited by computation capacity, display size, and visual understanding, the number of topics generated by flat models tends to be limited, and accordingly the topics are rather general and coarse-grained. On the other hand, hierarchical topic modeling offers better understanding of the data corpus by representing information at multiple levels of detail, and allowing people to interactively provide feedback at different aggregation levels. Using a hierarchy, users can explore high-level coarse topics and zoom in on fine-grained topics. Users can drill down into a subset of data to increase understanding (sense-making) and organize computed topics into a hierarchy that matches users' mental model. By focusing on steering unclear parts and leaving the rest to the computational methods, more efficient and comprehensive discovery is possible. In addition, flat topic modeling methods assume that all topics are at the same level, regardless of semantic granularity or size. For example, flat topics generated from sports articles during world cup season may contain many topics related to soccer but few topics on other sports. On the other hand, users may want to organize topics into various levels according to their mental models, e.g., by sports.

Despite the aforementioned advantages of hierarchical topic models over flat models, limited work has been done on interactive hierarchical topic modeling. A few visual analytics systems that support hierarchical topic modeling offer interactions to explore multiple levels of the underlying hierarchy, but their capability to modify or steer the underlying model is limited. For flexible steerability, we propose that the visualization systems, underlying computational algorithms, and users' interactions should be *tightly integrated*. Tight integration refers to the algorithm, visualization, and user interaction being jointly developed, where all three components are considered throughout the design process. Visualizations should not only show the outputs of models, but serve as the medium for interaction. Interaction should not be limited to controlling some parameters of algorithms, but allow higher-level operations that support the discovery process. Finally, algorithm development should take into account not only automated performance metrics, but consider interactivity and transparency for visualization in their formulation. This process of co-development goes beyond the adaptation of existing methods to meet the needs of users or interactive tasks, but instead co-designs algorithms, interactions, and visualizations simultaneously to ensure proper synchronization, compatibility, and performance. The proposed work explores the paradigm of tight integration and proposes a new way to implement tight integration for interactive hierarchical topic modeling.

In this paper, we present ArchiText, a visual analytic system using hier**Arch**ical **I**nteractive topic modeling for large-scale **Text** data. ArchiText visualizes hierarchical topics and offers various interactions to steer the topics and their hierarchical structure. ArchiText closely integrates the computational formulation of the model with the interactions provided to support flexible and rapid updates. The primary contributions of this work include:

- Development of an interactive and hierarchical topic modeling algorithms that achieve tight integration among visualization, computational model, and visual representation.
- Implementation of a visualization prototype system for large-scale document analysis utilizing our interactive hierarchical topic modeling framework.

2 Related Work

2.1 Visualization of Text Corpora

Organizing large, unstructured document collections into semantically meaningful topics and visualizing them has been a widely studied problem in the visualization community [5]. The earliest works including Topic Island [36] and IN-SPIRE [20] visualize document items and extracted themes. With the introduction of modern topic modeling methods such as Probabilistic Latent Semantic Analysis (PLSA) [21], Latent Dirichlet Allocation (LDA) [2], and Nonnegative Matrix Factorization (NMF) [31], research on text visualization has been substantially accelerated. Specifically, many results have focused on visualizing topic analysis results and allowing interactive exploration of topics and data items without the ability to steer the topic modeling results.

Once computed, topics are generally represented as a set of the most representative keywords. In many cases, the similarities between topics or documents are taken into account. Documents are represented as two-dimensional or three-dimensional points by applying dimension reduction techniques to the topic analysis results. In this way, similar documents and similar topics are placed closer to each other. For instance, UTOPIAN [8] maps documents into a 2D scatterplot, in which clusters are labeled with keywords, and users can interact with the topic modeling results. Other examples include iVisClustering [32] and TopicLens [28]. Several works such as ContexTour [34], FacetAtlas [7], SolarMap [6], and Concept Visualizer [19] adopt contours to represent static topics and relationships among them.

2.2 Interactions in Topic Modeling

Fully leveraging interactivity provided by visual analytics, several systems have incorporated a 'human-in-the-loop' approach to interactively modify the underlying topic model. Automatically generated topics often can be of low quality and noisy; or may not align well with user's mental model. In these cases, human-in-the-loop topic modeling techniques allow users to steer underlying topic models to obtain better results [25]. For example, if automatically generated topics contain two similar topics, users may want to merge them into a single topic. To this end, various interactions are introduced including *add, modify, split, combine, and remove* topics, documents representing the topics, and keywords [8,9,11,13,14,22,23,32,37,44]. We have surveyed interactive topic modeling systems with model steerability and organized the user interactions into word-level, document-level, and topic-level based on the unit of interactions. Specifically, the word-level interactions include user's activities of refining topics by adding words to a topic, moving words between topics, removing words from a topic, re-weighting word importance for a topic, and creating a new topic using selected words. Similarly, the document-level interactions involve user's editing of topics (which can be viewed as document clusters) by moving documents from one topic to another topic, removing documents from its parent

Unit	Interaction Tasks	Reasons/Goals	Prior Work
Word	Create a topic by seed word(s)	Need a new topic around the word(s).	[8, 14]
	Add word(s) to a topic	The word(s) is relevant to the topic.	[14, 44] ([33])
	Move word(s) from a topic to another.	The word(s) is more relevant to another topic.	
	Remove word(s) from a topic	The word(s) is not relevant to the topic.	[14, 44] ([33])
	Confirm or reject a word from a topic	The word(s) is definately relevant/irrelevant for the topic.	[9, 37]
	Change word distribution of a topic	The topic is better represented with new word distribution.	[8, 32, 44] ([33])
	Add word(s) to stopword list	The word(s) is not good representative of the data.	[37, 44]
Doc	Create a topic by seed document(s)	Need a new topic around the topic of the document(s).	[8]
	Move document(s) from a topic to another	The document(s) belongs to another topic.	[14, 32] ([33])
	Remove document(s) from a topic	The document(s) is irrelevant or of low quality.	[32, 44] ([33])
	Confirm or reject a document from a topic	The document(s) is relevant/irrelevant for the topic.	[13]
Topic	Merge two topics into a topic	Two topics are very similar.	[8, 11, 14, 22, 32, 44]
	Split a topic into sub-topics	The topic is too broad or not coherent.	[8, 14, 22, 32, 44] ([33])
	Move a topic	The topic belongs to another branch.	[11, 23, 32]
	Remove a topic	The topic is irrelevant, uninteresting, or of bad quality.	[14, 23, 32] ([33])
	Restore a topic	Need to undo 'remove topic'	
	Fix/freeze a topic	The topic is final and no more refinement is needed.	
	Collapse topics (show fine-grained)	Topics are too general; there are not enough topics.	[23]
	Aggregate topics (show coarse-grained)	Topics are too specific; there are too many topics.	[23]

Table 1: User interaction tasks for model steering supported (or suggested) in previous works. Parenthesis indicates suggested interaction tasks.

topic, re-weighting document importance for a topic, and creating a new topic using selected words. The topic-level interactions occur when users perform group-wise interactions such as merging, splitting and removing topics. A complete list of user interactions that are available in some of the existing interactive topic model systems is summarized in Table 1. However, user interactions in many prior works have been designed for algorithmic convenience rather than user tasks [33], and thus are not tightly coupled with the underlying algorithms.

2.3 Visual Analytics for Hierarchical Topic Modeling

In a number of recent papers, the topics are organized with a hierarchy. Hierarchical topic modeling and hierarchical document clustering techniques organize documents into various granularity of topics. In this way, large text corpora can be analyzed and understood through multi-scale analysis. In hierarchical visual analytics systems, users interactively navigate through topic hierarchy visualizations to find coarser grained (higher nodes) or finer grained (lower nodes) topics. For instance, HiPP [41] uses a hierarchical circle packing algorithm, in which a topic or a document is represented as a circle. Topic circles can be expanded into sub-topic circles down to individual documents. Other systems directly visualize a topic tree using node-link style visualizations. For example, Brehmer et al. [3] introduces Overview, a visual document mining tool for investigative journalists. Overview allows users to explore the topic hierarchy and annotate relevant documents for later use. Similarly, Dou et al. [11] visualize topics and their temporal patterns as tree and themeriver charts, respectively. Other works focus on the evolution of topic hierarchies [10, 35] and matching topic hierarchies from multiple sources [48]. A work more closely related to what we propose here is by Hoque and Carenini [23] which utilizes a simple collapsible tree in an online conversation analytics system and allows users to explore and revise a topic hierarchy by moving topic nodes. Sunburst visualizations [46] are also used to visualize topic hierarchies [43, 49], where concentric circles represent different levels of the topic hierarchy starting from the the center (source node) to the outermost (leaf nodes). This technique relies strongly on user interaction to allow users to expand sub-components of the tree if requested.

Among these hierarchical topic modeling systems, only a few support an interactive modification of the underlying hierarchical model [11, 23]. These systems only offer group-wise or topic-level organizational operations such as merging topics, splitting a topic, and moving a topic under a new parent. Therefore, users are unable to steer the underlying topic model to a finer degree (e.g., by words and/or documents). More recently, IHTM [14] proposes a mixed-initiative approach where a human can intervene during the incremental model building process. Users are asked to choose from several interaction strategies with the help of a preview of the expected outcome of each strategy. While this work has many advantages that work well for very small datasets, it is not scalable to large-scale datasets because the underlying topic hierarchy is optimized every time a data item is entered. Also, the interaction strategies available in the IHTM system are limited and its word-level interactions are offered only before the algorithm starts.

2.4 Interactive Model Steering in Visual Analytics

Mixed-Initiative systems [24], which combine human knowledge and human intelligence to create a collaborative system between users and machines, are closely related to what we propose in this paper. Adopting this principle, in mixed-initiative visual analytics systems, users interact with the machine via visual interfaces to *steer the model* by controlling different model parameters. Some systems offer direct manipulation of model parameters through control panels. However, direct manipulation of model parameters requires a deep understanding of the underlying model mechanism and its parameters. Endert et al. [16] introduce 'semantic interaction' to steer the models using native user interactions on visual objects rather than model parameters. For instance, Disfunction [4] and Observation-Level Interaction [17] allow users to move points in a 2D scatterplot to update the underlying distance function. Podium [47] updates an SVM model as users change the order of data items. Other examples include [15, 26, 30, 38].

3 Interactive Topic Modeling with Tight Integration

In this section, we propose our novel approaches for interactive hierarchical topic modeling. We first identify design goals for tight integration in interactive hierarchical topic modeling. Then we propose our modular interaction design to support flexible user feedback. Finally, we describe the underlying algorithms for base operations and interaction tasks.

3.1 Design Goals for Tight Integration

Tight integration advocates for the visualization accurately representing the computational result with reasonable responsiveness, user interaction being accurately interpreted taking advantage of more detailed information that the underlying algorithm offers, and flexibility in the model to accommodate a wide range of user tasks and goals. Here, we list the design goals of tight integration and how ArchiText achieves them.

Fast, Adaptive, and Interaction-conductive Model and Algorithm. The foundational algorithm should be designed and developed

with user interaction considered from the start. Tight integration synchronizes updates in the underlying computation with the interpretation of the user interaction. This update cycle is iterative, where the underlying computational methods guide the changes in the intermediate results taking the user interaction into account. To achieve fast and accurate visualization updates, the underlying updates of the computational result should not involve recomputing the solution from scratch. Rather, underlying computations should be tailored to allow incremental, timely, and responsive updates. In our proposed system, results are adaptively updated based on intermediate solutions and user refinement.

Visualization of Various Degrees of Information and User Feedback. The computational results, the internal factors, and characteristics of the algorithms should be exposed in various degrees of information level, likely through multiscale visual representations. Interactively, users may perform operations on specific information, yet the algorithmic interpretation of the action will need to consider additional information. For instance, removing a topic is likely based on a subset of keywords for a topic shown in the visualization. However, the underlying algorithm contains many additional details about the topic, such as the complete keyword distribution (what a topic is) and the topic distribution for each document (how close a document is to a topic). Without careful interpretation of user intention incorporated into algorithms, the results can quickly be distorted after multiple interactions because of the limited information users are shown in the visualization. In addition, with a hierarchical topic modeling, visualizations can show multiple levels of detail, aggregating or de-aggregating sub-hierarchies depending on the level of detail requested by users.

Capability to Support a Variety of User Feedback Types. The computational models and algorithms should be flexible enough to incorporate various user intentions and tasks. Various model steering interactions have been identified as important (Table 1), but not all interactions were supported in a single system previously. One reason is that most topic modeling methods have many parameters and settings that are difficult to properly tune to produce the results that meet user expectations. We chose Nonnegative Matrix Factorization (NMF) [31] as our underlying foundational topic modeling method due to its flexibility and efficiency. Multiple advantages of NMF that we have observed and analyzed in our previous work [8,12,27] include fast algorithms, higher quality and more consistent solutions, flexibility to changes in tasks, adaptive updating methods [1,40,50], and interpretability of results. In addition, since interactions can be formulated as constrained NMF problems, we can identify a set of primitives that are common over various interaction tasks and build core computational modules that can be utilized across them as will be described in the next section. These important features of our algorithm combine to facilitate our tight integration methodology.

3.2 Interaction Primitives for Hierarchy Steering

To achieve the goal of the tight integration, we propose to break down a large suite of interaction tasks into basic operations called primitives. These primitive operations can be optimized individually and then combined to implement specic subtasks. Surveying interaction tasks supported or suggested in existing works, we observed that all interaction tasks can be further divided into tree operations and/or supervised topic computation. For instance, *Move a topic into a new parent* (**MoveT**) can be achieved by 'cut out a topic' followed by 'insert the topic under a new parent' with re-computations. Based on interaction tasks supported in existing systems and our design goals, we come up with five base operations, whose combination can form the interaction tasks in Table 1. This modular implementation makes it possible to optimize the tightly coupled system performance by fine tuning the five base operations. The five base operations are as follows:

1. **makechildren(T)**: Create two child topics for a leaf node topic T
2. **merge(T1, T2)**: Merge sibling topic nodes T1 and T2
3. **insert(T1, T2)**: Insert a topic node T1 under a new parent node T2
4. **cut(T1)**: Cut out a topic node T1
5. **recompute(T)**: Recompute child topics of T using a constrained NMF topic model

Notation	Description
T_i	The i-th topic node
$D(T_i)$	Indices of documents that belong to T_i
$p(T_i)$	The parent topic node of T_i
C	Trash can, i.e., the set of removed topics
m	The number of keywords
n	The number of documents
n_i	The number of documents in T_i
k_i	The number of child topics under T_i
$X^{(i)}$	The $m \times n_i$ word-document matrix of T_i
$W^{(i)}$	The $m \times k_i$ word-topic matrix of child topics of T_i
$w_p^{(i)}$	The p-th column of $W^{(i)}$
$H^{(i)}$	The $k_i \times n_i$ topic-document matrix of child topics of T_i
$h_q^{(i)}$	The q-th column of $H^{(i)}$
$\mathbb{R}^+$	The set of nonnegative real numbers
$\|\|\cdot\|\|_F$	The Frobenius norm
$A_{r\cdot}$	The r-th row of matrix A
$A_{\cdot r}$	The r-th column of matrix A
$argmax(a)$	The index of the largest element in vector a

Table 2: Key notations used in the paper.

Utilizing the base operations, we simultaneously design user interactions and corresponding algorithms. The complete list of supported interaction tasks is described in Table 3 and Section 3.4.3.

3.3 NMF for Topic Modeling

We define our notation and topic modeling formulations as follows. Conceptually, a topic T is identified as a keyword distribution and has a set of documents that belong to the topic (i.e., the documents whose topic distribution has the strongest weight in the topic). A topic node in the topic hierarchy is denoted as $(T, D, p(T))$ where T denotes the topic, D the set of documents in the topic, and $p(T)$ a reference to the parent topic of the topic T. We denote the i-th topic as T_i and the indices of documents that belong to T_i as $D(T_i) = \{d_{i_1}, \cdots, d_{i_{n_i}}\}$, where n_i is the number of documents in the topic. We reference the parent topic node of a child topic T_i as $p(T_i)$. Note that the documents in a topic are the union of documents that belong to its child topics, i.e., $D(T_i) = \cup_{p(T_j)=T_i} D(T_j)$.

Let $X \in \mathbb{R}_+^{m \times n}$ be the data matrix of topic T, where m is the number of words in the corpus and n is the number of documents that belong to the topic. The p-th column of X represents the bag-of-words representation of document d_p with respect to m keywords. A standard Nonnegative Matrix Factorization (NMF) approach solves a low-rank approximation as follows:

$$\min_{\{W,H\} \geq 0} ||X - WH||_F^2, \tag{1}$$

where $W \in \mathbb{R}_+^{m \times k}$ and $H \in \mathbb{R}_+^{k \times n}$ are factor matrices and k is the number of child topics under T. W describes topics and H describes document-topic memberships. The p-th child topic under T is calculated as w_p, the p-th column of W. High values in w_p indicate that the corresponding words are strongly associated with the p-th child topic. Next, the q-th column of H, h_q, represents document d_q as a weighted combination of k topics. We say document d_q belongs to the p-th child topic if the p-th element of h_q is its largest element, i.e., $p = argmax(h_q)$. Notations are summarized in Table 2.

In order to steer topics, we modify the NMF formulation as follows:

$$\min_{\{W,H,U\} \geq 0} ||X - WH||_F^2 + \alpha||M_W \circ (W - W')||_F^2 + \beta||M_H \circ (E - HU)||_F^2 \tag{2}$$

The second term influences topics' keyword descriptions by forcing W to be similar to W', which represents topic keywords selected by the users. The third term affects topic-document memberships by forcing H to be similar to E, which represents topic-document memberships assigned by the users, with the help of a scaling matrix U. Parameters α and β determines the amount of user control for word-level and document-level interactions, respectively. When $\alpha = 0$ (or $\beta = 0$), the word- (or document-) level interaction is not reflected into the model. Larger α, β leads to stronger incorporation of user steering, but

it may result in less truthful representation of the underlying data. α, β are set to be proportional to the number of interacted topics by word-level and document-level tasks, respectively. M_W, M_H are masking matrices where $(M_W)_{\cdot r} = 0$ and $(M_H)_{r\cdot} = 0$ for $r \notin \{\text{steered indexes}\}$ and $(M_W)_{\cdot r} = 1$ and $(M_H)_{r\cdot} = 1$ for $r \in \{\text{steered indexes}\}$. More detail will be described in Section 3.4.3.

3.4 Algorithm

3.4.1 Base Operations

In this section, we describe the algorithms for our base operations based on which our interaction tasks can be composed.

$[T_1, T_2] =$ **makechildren**(T_0)

---

makechildren applies a rank-2 NMF algorithm to the documents in a topic node T_0 to create two of its children nodes T_1, T_2.

Solve $\min_{\{W^{(0)},H^{(0)}\}\geq 0} ||X^{(0)} - W^{(0)}H^{(0)}||_F^2$ where $W^{(0)} \in \mathbb{R}_+^{m\times 2}$, $H^{(0)} \in \mathbb{R}_+^{2\times n_0}$

$D(T_k) = \{d_q \in D(T_0) | argmax(H^{(0)}_{\cdot q}) = k\}$, $p(T_k) = T_0$ for $k = 1, 2$

$T_0 =$ **merge**(T_1, T_2)

---

merge creates a new parent T_0, which is the union of selected sibling topic nodes T_1, T_2, under their original parent $p(T_1)$.

$p(T_0) = p(T_1)$ and $D(T_0) = D(T_1) \cup D(T_2)$
$p(T_k) = p(T_0)$ for $k = 1, 2$

---

insert(T_1, T_2)

---

insert adds a topic T_1 under the selected node T_2 and updates its ancestors.

$p(T_1) = T_2$ and $D(T_2) = D(T_1) \cup D(T_2)$
$parent = p(T_2)$
while parent is not the top node **do**
  $D(parent) = D(parent) \cup D(T_1)$
  $parent = p(parent)$
end while

---

cut(T_1)

---

cut removes a topic T_1 from its ancestors.

$parent = p(T_1)$
$p(T_1) = null$ and $D(p(T_1)) = D(p(T_1)) \backslash D(T_1)$
while parent is not the top node **do**
  $parent = p(parent)$
  $D(parent) = D(parent) \backslash D(T_1)$
end while

$[T'_{01}, \cdots, T'_{0k_0}] =$ **recompute**$(T_0, (W'), (H'))$

---

recompute applies a flat NMF algorithm with constraints on a topic T_0 to re-partition its children $T'_{01}, \cdots, T'_{0k_0}$. The second and third terms incorporate word-level and document-level supervision, respectively.

Solve $\min_{\{W^{(0)},H^{(0)},U^{(0)}\}\geq 0} ||X^{(0)} - W^{(0)}H^{(0)}||_F^2 + \alpha||M_W^{(0)} \circ (W^{(0)} - W'^{(0)})||_F^2 + \beta||M_H^{(0)} \circ (E^{(0)} - H^{(0)}U^{(0)})||_F^2$ where $W^{(0)} \in \mathbb{R}_+^{m\times k_0}$, $H^{(0)} \in \mathbb{R}_+^{k_0\times n_0}$, $U^{(0)} \in \mathbb{R}_+^{k_0\times k_0}$

$D(T_k) = \{d_q \in D(T_0) | argmax(H^{(0)}_{\cdot q}) = k\}$ for $k = 1, \cdots, k_0$

When performing **recompute** on a parent topic before **cut** on a child topic, we redistribute documents that are not strongly relevant to the child topic (i.e., $max(H_{kq}) < threshold$) into their sibling topics. As a result, when moving or removing topics, keywords, or documents, only the documents that are strongly relevant to the moved or removed topics, keywords, or documents are cut out.

3.4.2 Hierarchical Topic Initialization

The proposed system generates the initial hierarchical topics where the upper level topics are more general and larger, and the lower level topics are more specific, finer-grained, and more tightly related. We adopted a hierarchical topic modeling algorithm called HierNMF2 [18, 29], which uses a fast rank-2 NMF [12] and a binary tree splitting rule. In other words, we recursively split a topic by solving Eqn. 1 with $k = 2$ topics. By utilizing the simple computation to obtain a rank-2 NMF, some very substantial speedups have been achieved for computing topic modeling results, which can be highly beneficial for achieving real-time interaction.

3.4.3 Hierarchical Topic Revision

After the initial hierarchical topics are computed, users can steer the model by performing various tasks described in Table 3. We grouped user tasks by interaction unit types: topics, words, and documents. Note that previous hierarchical topic modeling systems allow topic reorganization through some topic-level interactions, but they offer none to highly limited word-level or document-level topic steering.

Topic-level Tasks

Topic-level tasks in existing hierarchical topic modeling systems affect all documents in the interacted topic. For example, moving a topic would relocate all the associated documents into a new parent topic. However, the decisions to do so are based on limited information shown on the screen to the users. Thus, our approach does not move all documents of a topic when moving/removing topics, but rather moves only a strongly relevant subset using constrained NMF. Now we define topic-level tasks as follows:

$T_0 =$ **MergeT**(T_1, T_2)

MergeT combines selected topics T_1 and T_2 to create a new parent topic T_0. If T_1 and T_2 are not siblings, T_1 is moved under T_2's parent before merging.

if $p(T_1) \neq p(T_2)$ **then**
  MoveT$(T_1, p(T_2))$
end if
$T_0 =$ **merge**(T_1, T_2)

$[T_1, T_2] =$ **SplitT**(T_0)

SplitT partitions topic T_0 into two child topics T_1 and T_2 using rank-2 NMF.

$[T_1, T_2] =$ **makechildren**(T_0)

MoveT(T_1, T_2)

MoveT detaches a topic T_1 and attaches it under a new parent T_2. Before detaching T_1, we redistribute less relevant documents in T_1 into its sibling topics to move only a strongly relevant subset. After attaching, we solve another NMF for the new parent T_2 to find more suitable child topics with the incoming topic T_1.

recompute$(p(T_1), w_1)$ with $\min_{\{W,H\}\geq 0} ||X - WH||_F^2 + \alpha||M_W \circ (W - W')||_F^2$ where $W'_{\cdot 1} = w_1$.
cut(T_1)
insert(T_1, T_2)
recompute$(p(T_2))$ with $\min_{\{W,H\}\geq 0} ||X - WH||_F^2$

RemoveT(T_1)

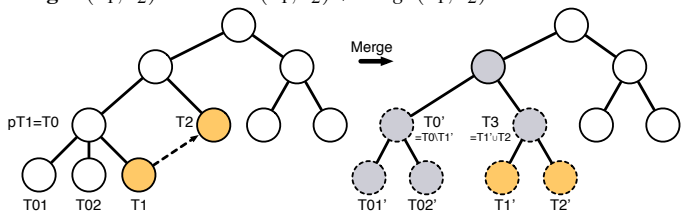
RemoveT discards the most relevant documents of the selected topic T_1 into the trash C rather than all documents in T_1. The remaining less-relevant documents are redistributed to its sibling topics.

recompute$(p(T_1), w_1)$ with $\min_{\{W,H\}\geq 0} ||X - WH||_F^2 + \alpha||M_W \circ (W - W')||_F^2$ where $W'_{\cdot 1} = w_1$.
cut(T_1)
insert(T_1, C)

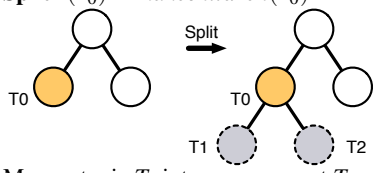
Diagram Operated Affected Changed

Topic-level

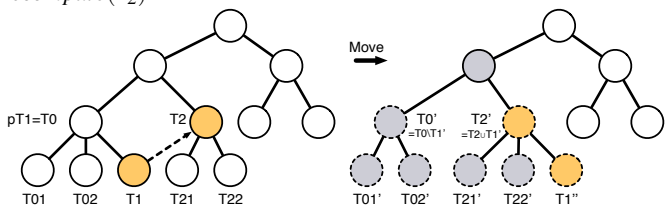
Merge two topics T_1 and T_2
MergeT(T_1,T_2) = **MoveT**(T_1,T_2) + $merge(T_1,T_2)$



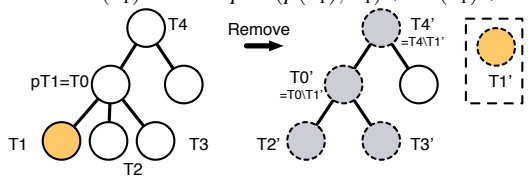
Split a topic T_0 into child topics
SplitT(T_0) = $makechildren(T_0)$



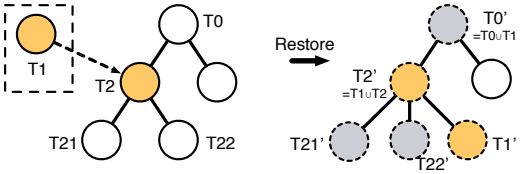
Move a topic T_1 into a new parent T_2
MoveT(T_1,T_2) = $recompute(p(T_1),w_1)$ + $cut(T_1)$ + $insert(T_1,T_2)$ + $recompute(T_2)$



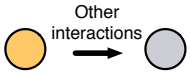
Remove a topic T_1
RemoveT(T_1) = $recompute(p(T_1),w_1)$ + $cut(T_1)$ + $insert(T_1,C)$



Restore a topic T_1 under T_2 from trash
RestoreT(T_1,T_2) = $cut(T_1)$ + $insert(T_1,T_2)$ + $recompute(T_2)$



Fix a topic T_1
FixT(T_1): makes T_1 unaffected by $recompute(p(T_1))$



Word-level

Create a topic T_w by seed words w under a parent T_1
CreateTW(w,T_1) = $recompute(T_1,w)$

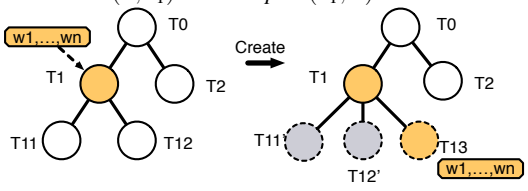
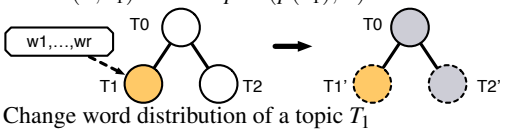


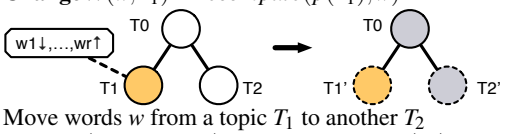
Table 3: User Interactions Supported by ArchiText.

Word-level (Cont'd)

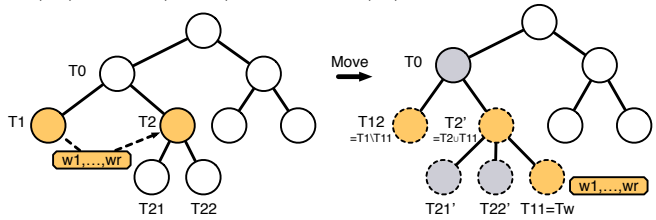
Add words w to a topic T_1
AddW(w,T_1) = $recompute(p(T_1),w)$



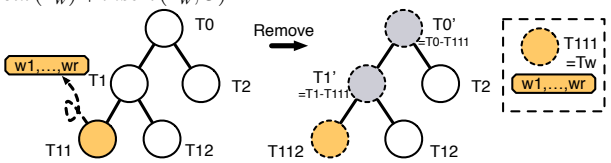
Change word distribution of a topic T_1
ChangeW(w,T_1) = $recompute(p(T_1),w)$



Move words w from a topic T_1 to another T_2
MoveW$(w \in T_1,T_2)$ = $makechildren(T_1)$ + $recompute(T_1,w)$ + $cut(T_w)$ + $insert(T_w,T_2)$ + $recompute(T_2)$

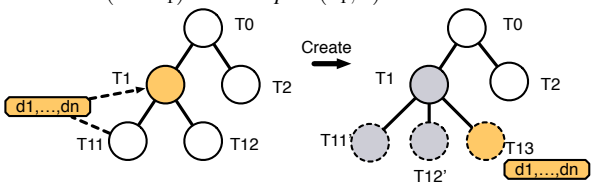


Remove words w from a topic T_{11}
RemoveW(w,T_{11}) = $makechildren(T_{11})$ + $recompute(T_{11},w)$ + $cut(T_w)$ + $insert(T_w,C)$

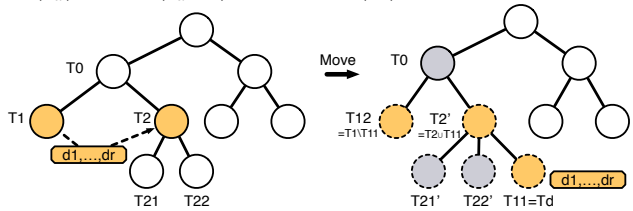


Document-level

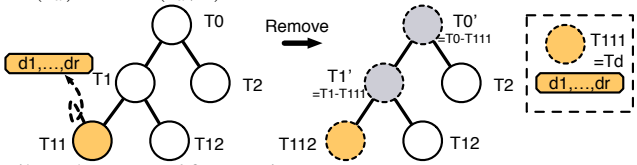
Create a topic T_d by seed docs d under a parent T_1
CreateTD$(d \in T_1)$ = $recompute(T_1,d)$



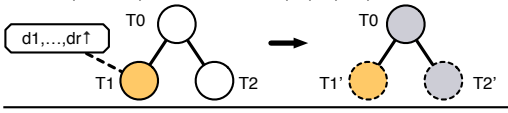
Move documents d from a topic T_1 to another T_2
MoveD$(d \in D(T_1),T_2)$ = $makechildren(T_1)$ + $recompute(T_1,d)$ + $cut(T_d)$ + $insert(T_d,T_2)$ + $recompute(T_2)$



Remove documents d from a topic T_{11}
RemoveD(d,T_{11}) = $makechildren(T_{11})$ + $recompute(T_{11},d)$ + $cut(T_d)$ + $insert(T_d,C)$



Like a document d for a topic T_1
LikeD$(d \in T_1)$ = $recompute(p(T_1),d)$



RestoreT(T_1, T_2)

RestoreT moves a previously deleted topic T_1 from the trash C into a selected parent topic T_2.

cut(T_1)
insert(T_1, T_2)
recompute$(p(T_2))$ with $\min_{\{W,H\}\geq 0} ||X - WH||_F^2$

Fix(T_1)

Fix freezes a topic T_1 so that it will not be changed in later stages. It can be used when the quality of a current topic is determined good and the user wants to consider the topic as final. The **Fix** task does not involve any computation but marks the topic as fixed so that the topic is not modified in any subsequent computation or interaction.

Word-level Tasks

Word-level interactions influence keyword descriptions of topics so that a specific topic becomes more (or less) about certain words. As a result, some documents are redistributed according to the new topic descriptions. Previously, word-level interactions were only supported in some flat topic model systems but not in hierarchical systems. To our knowledge, ArchiText is the first to allow word-level refinement of hierarchical topics. For simplicity and efficiency, we limit the scope of affected topics to one level. We define word-level tasks as follows:

CreateTW(w, T_1)

CreateTW creates a new topic with seed words w under a parent topic T_1.

$k_1 = k_1 + 1$
recompute$(p(T_1), w)$ with $\min_{\{W,H\}\geq 0} ||X - WH||_F^2 + \alpha||M_W \circ (W - W')||_F^2$ where $W'_{\cdot 1} = w$.

AddW(w, T_1)

AddW adds new terms w to a topic T_1 to steer it toward the selected words.

recompute$(p(T_1), w)$ with $\min_{\{W,H\}\geq 0} ||X - WH||_F^2 + \alpha||M_W \circ (W - W')||_F^2$ where $W'_{\cdot 1} = w$.

ChangeW(w, T_1)

ChangeW changes the word distribution w of a topic T_1 to steer the topic based on the re-weighted words..

recompute$(p(T_1), w)$ with $\min_{\{W,H\}\geq 0} ||X - WH||_F^2 + \alpha||M_W \circ (W - W')||_F^2$ where $W'_{\cdot 1} = w$.

MoveW$(w \in T_1, T_2)$

MoveW aims to subtract the selected terms w from a topic T_1 and add them into another topic T_2 by moving the most relevant documents in the topic.

$[T_w, T_{\bar{w}}]$ = **makechildren**(T_1) with $\min_{\{W,H\}\geq 0} ||X - WH||_F^2 + \alpha||M_W \circ (W - W')||_F^2$ where $W'_{\cdot 1} = w$.
MoveT(T_w, T_2)

RemoveW(w, T_1)

RemoveW discards the documents in a topic T_1 that are most relevant to the selected words w. The remaining less relevant documents are redistributed to sibling topics.

$[T_w, T_{\bar{w}}]$ = **makechildren**(T_1) with $\min_{\{W,H\}\geq 0} ||X - WH||_F^2 + \alpha||M_W \circ (W - W')||_F^2$ where $W'_{\cdot 1} = w$.
RemoveT(T_w)

Document-level Tasks

Document-level interactions influence document-topic memberships to steer a topic to be similar (or dissimilar) to the selected documents. As a result, keyword descriptions of affected topics can change accordingly. Following the tight integration principles, our document-level tasks not only involve the selected documents, but also affect documents that are similar or relevant to the selected documents. We define document-level tasks as follows:

CreateTD(d, T_1)

CreateTD creates a new topic with seed document d under a parent topic T_1.

$k_1 = k_1 + 1$
recompute$(p(T_1), d)$ with $\min_{\{W,H\}\geq 0} ||X - WH||_F^2 + \beta||M_H \circ (E - HU)||_F^2$ where $E_{1d} = 1$.

MoveD$(d \in T_1, T_2)$

MoveD aims to subtract the selected documents d (and similar ones) from a topic T_1 and add them into another topic T_2.

$[T_d, T_{\bar{d}}]$ = **makechildren**(T_1) with $\min_{\{W,H\}\geq 0} ||X - WH||_F^2 + \beta||M_H \circ (E - HU)||_F^2$ where $E_{1d} = 1$.
MoveT(T_d, T_2)

RemoveD(d, T_1)

RemoveD discards the selected documents d and the similar ones from a topic T_1. The remaining less-relevant documents are redistributed to sibling topics.

$[T_d, T_{\bar{d}}]$ = **makechildren**(T_1) with $\min_{\{W,H\}\geq 0} ||X - WH||_F^2 + \beta||M_H \circ (E - HU)||_F^2$ where $E_{1d} = 1$.
RemoveT(T_d)

LikeD$(d \in T_1)$

LikeD steers a topic T_1 to be more like the liked document d.

recompute$(p(T_1))$ with $\min_{\{W,H\}\geq 0} ||X - WH||_F^2 + \beta||M_H \circ (E - HU)||_F^2$ where $E_{1d} = 1$.

4 System

In this section, we describe ArchiText, our prototype visual analytics system for interactive hierarchical topic modeling with tight integration. The proposed system is built using the D3.js visualization library, Flask framework, sqlite database, and a fast rank-2 nonnegative matrix factorization algorithm and the proposed constrained low rank approximation shown in Eqn. 2 written in Matlab™.

4.1 System Design

Our system has two modes: a topic workspace mode (Fig. 1) and a hierarchy view mode (Figs. 2-3). The topic workspace mode is designed to facilitate flexible user interactions for tuning and interacting with the hierarchical topic representation. The hierarchy view mode is primarily for inspecting the overall structure of the computed topic tree. Users can alternate between the two modes by clicking the blue button on the top right corner shown in Fig. 1a.

Topic Workspace Mode contains the main topic card view, a control bar, a breadcrumb view, and a mini overview.

The main topic card view (Fig. 1c) visualizes topics up to selected depths in the computed hierarchical topic tree. Each topic is visualized as an indented equal-width card where the height of each card is proportional to the number of documents that belong to the topic. Each topic's most representative keywords and their importance weights are visualized as a sorted list with bars. This design allows users to quickly understand topics well [45] and easily compare keyword weights across

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2020.2981456, IEEE Transactions on Visualization and Computer Graphics

Fig. 2: The hierarchy view mode presents a high-level overview of the topic tree with a control panel. A topic can be expanded (or collapsed) to show (or hide) its child topics. Expandable topics are represented as filled circles.

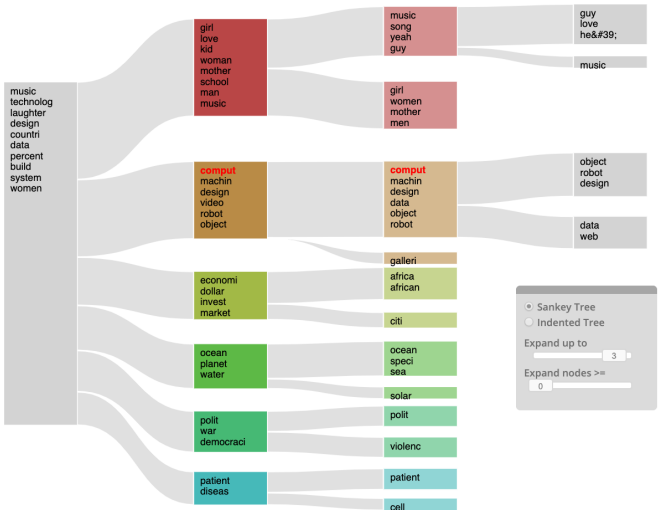


Fig. 3: An alternate Sankey tree visualization in the hierarchy view mode. Placing the mouse over a keyword highlights the keyword in other topics.

different topics. Note that all keywords are in stemmed forms as we use the porter stemmer during the data preprocessing step. On the top right corner of a topic card are the menu button and the flip button. A topic card can be flipped to show its documents (Fig. 1d) with their 'Thumbs up' buttons to like the corresponding documents. Hovering the mouse over a document shows its detailed information such as document id, title, authors, etc (Fig. 1e). Topic coherence scores are visualized as bars on top of the topic cards. As a topic coherence metric, we use pointwise mutual information (PMI), which is highly correlated with human judgement [39]. This can guide users to focus on refining and improving low-quality topics and observe how their interactions affect topic quality by monitoring the bars. Topic cards, keywords, and documents can be interacted with to steer the underlying topics and their hierarchy, which will be described in Section 4.2 in detail.

The control bar contains buttons to update the main topic card view. A sliding trash panel is toggled by clicking a trash button in the control bar (Fig. 1a). The plus and minus buttons in the control bar (Fig. 1a) change the visualized depth of the topic tree to support multi-level exploration. Users can 'drill down/zoom in' on a topic for more details and 'zoom out' to see higher-level topics as indented topic cards, respectively. For example, Fig. 4a shows the first level topics. By zooming in, Fig. 4b shows the first level and the second level topics where the deepest visible topic card is shown and the rest are collapsed. By zooming in again, Fig. 4c shows topics up to the third level. Note that parent and children topics have the same color hue, but with different saturation; lighter colors and longer indentations represent deeper node depths.

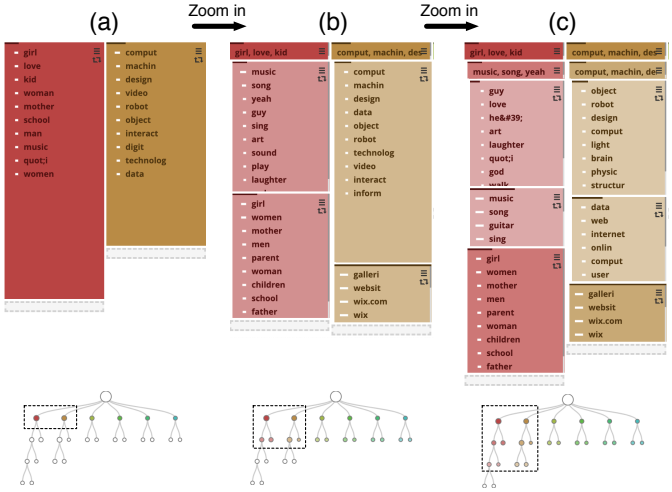


Fig. 4: Zooming-in from (a) to (c) by clicking the + button reveals deeper levels of hierarchical topics interactively.

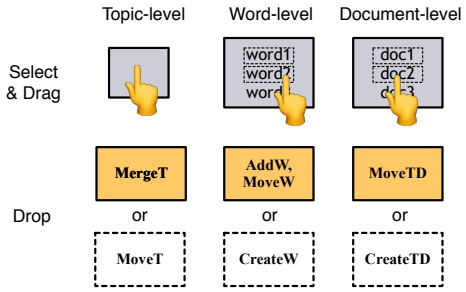


Fig. 5: User interaction design for supported tasks.

The mini overview displays the overall topic hierarchy as either a weighted tree (Fig. 1f) or an icicle visualization. Topics visualized as topic cards are colored accordingly and the remaining topics with deeper depths are colored gray. When hovering over a topic, the breadcrumb view (Fig. 1b) shows the trail from the top node to the current node (orange-colored topic in Fig. 1c), and the corresponding node in the mini overview is highlighted with black solid line (orange-colored circle in Fig. 1f).

Hierarchy View Mode offers two types of tree visualizations, an indented tree (Fig. 2) and a Sankey tree (Fig. 3), which can be selected in the control panel (Fig. 2a). In both views, topic colors correspond to those in the workspace mode. Using the control panel (Fig. 2a), users can collapse topics by their depths or sizes. The indented tree view (Fig. 2) visualizes the topics and the hierarchy as an indented tree where indentation reflects tree depth. For each topic, its ID, topic size, and top ten keywords are shown. Clicking a topic's circle collapses/expands the topic to hide/show its children topics. Filled circles represent collapsed (and thus expandable) topics. The Sankey tree view (Fig. 3) visualizes the topic tree from left (top node) to right (deeper level node) where node height reflects topic size (number of documents in a topic). Each topic node displays up to ten keywords depending on its size. Hovering over a topic node pops up a detail view showing all ten keywords and the size of a topic. When hovering over a keyword in a topic node, the same keywords in other topics are highlighted to show term patterns (red keywords in Fig. 3).

4.2 User Interaction Design

Fig. 5 demonstrates supported interaction tasks to steer the underlying hierarchical topics. All interactions are designed to be executed by clicking buttons (**SplitT**, **FixT**, **LikeD**) or simply dragging and dropping visual components such as words, documents, and topic cards as follows. When modifying existing topics (**MergeT**, **AddW**, **MoveW**, **MoveD**), drop recipients are the topic cards being modified. When cre-

ating a new topic (**CreateTW**, **CreateTD**) or moving a topic (**MoveT**, **RestoreT**), the drop recipient is a dotted space which represents a temporary topic card. When deleting words (**RemoveW**), documents (**RemoveD**), or a topic (**RemoveT**), the drop recipient is the trash button.

4.2.1 Interaction Assistant

In the proposed tight integration framework, users incrementally update models through a wide variety of interactions. During the process, our system guides users by predicting and recommending interaction tasks. Many of the interactions in our system start with selecting and dragging and end with dropping. Our interaction assistant is triggered when the user selects or drags something and then it predicts the next step to complete the interaction. This can be beneficial for users who are exploring potential alternatives for how to organize and construct their topics.

Multi-selection: When steering a topic using word-level (**CreateTW, AddW, MoveW, RemoveW**) or document-level (**CreateTD, MoveD, RemoveD**) interaction tasks, selecting multiple words or documents can convey clearer meaning than selecting a single word or a document. However, going over many words or documents can be time-consuming. To foster efficient multi-selection, our interaction assistant visually recommends selection candidates. When a word (or a document) is first selected, the system highlights frequently co-occurring words (or similar documents) in the same topic to be selected along with the first selected word (or document).

Drop: After selecting the words, documents, or a topic, the next step is to drag and drop them into another topic or into the trash. Our interaction assistant predicts and recommends the locations to drop them during an interaction. When the user starts dragging a topic, the system highlights similar topics as drop recipients to foster **MergeT** or **MoveT** tasks. Similarly, when the user starts dragging words or documents, the system highlights topics that are similar to the dragged words or documents as drop recipients to foster **AddW, MoveW, MoveD** or **CreateTW, CreateTD** tasks. If the selected words or documents are not coherent (not similar to each other), the system highlights the trash to foster **RemoveW, RemoveD** tasks. In addition, users can preview the expected hierarchy in the mini overview while placing the mouse over the drop recipients.

5 Experiments

In this section, we present a quantitative evaluation to show the scalability of our approach. For our experiments, we use a patent dataset[1]. This dataset contains about 7 million granted patents and their information, e.g., ID, type, title, abstract, year, etc. After filtering out non-utility patents, we are left with 6,248,456 utility patents.

We report computation time using different sizes of patent subsets. We select 10,000, 50,000, 100,000, 500,000, and 1,000,000 data items from the patent dataset to create multiple subsets of different sizes. For each subset, we report the running time of building the initial hierarchical topics as well as performing an interaction task on a topic that contains about 10% of the documents. Experiments were performed on a MacBook Pro with Intel Core i7 3GHz, 4 cores, 8GB memory. Table 4 shows that building an initial topic model with ten leaf nodes from a million documents takes about 5 minutes. Also, most tasks are finished within several seconds, supporting accurate and timely visualization in our tight integration methodology as discussed in Section 3.1.

In general, interactions with recomputations such as moving take longer computation time than interactions without recomputation such as merging (siblings) or splitting. This is because recomputation on a topic runs a flat NMF algorithm with constraints (Eqn. 2) on its parent topic, and the changes are propagated to its descendants. To reduce interaction latency caused by recomputation time, we suggest the following strategies. First, we could reduce the number of recomputation. Instead of performing full recomputation every time, the system can decide when to skip or perform recomputation. For instance, in the case of **recompute** before **cut** or after **insert**, recomputation can be skipped unless a large portion of the interacted topic is changed or a certain number of interaction tasks have been applied to the interacted topic without recomputation. Second, we could limit the number of iterations when solving Eqn. 2. Since our algorithm utilizes previous topics to initialize factor matrices W, H, we could reduce the number of total iterations per one recomputation and still reach a near optimal solution. Next, we can recommend users to keep the size of interaction small since recomputation time depends on the size of its parent topic. That is, focus on splitting topics into smaller ones rather than directly steering bigger topics. Because our recomputation is local, it only affects siblings. Regardless of the size of the entire dataset, users can use this strategy to achieve fast interaction.

Datasets	p10K	p50K	p100K	p500K	p1M
# Documents	10,000	49,995	99,989	499,968	999,941
# Words	6,585	15,414	22,702	60,131	92,966
Initialization	3.64	13.73	27.35	142.19	300.01
Merge	0.005	0.014	0.025	0.097	0.191
Split	0.066	0.138	0.308	1.164	7.009
Move	0.598	1.515	2.927	22.133	49.735
Move (w/o **re**)	0.023	0.036	0.050	0.186	0.356

Table 4: Computation times (in seconds) for hierarchy initialization with ten leaf nodes and several interaction tasks (merging siblings, splitting a topic, moving a topic–with and without recomputation). All results are averaged over 10 runs.

In this section, we do not report topic quality measures as those depend on user decisions of which topics to interact with. For instance, merging any two random topics would degrade the overall quality of the topic hierarchy. Instead, we show two use cases that showcase the effectiveness of our tight integration approach in Section 6.

6 Use Cases

6.1 TED Transcript Dataset

TED is a nonprofit organization that hosts influencial talks and shares the videos online. Various topics including technology, education, and self-help are covered in TED talks. Although the official TED website provides keyword search functionality and over 400 category tags, navigating about 3,000 talks and discovering talks of interest is not easy. In this section, we use ArchiText to understand main themes of the talks and organize them into hierarchical categories for easier navigation. We used the TED dataset containing 2,969 talk transcripts.[2]

A user starts by inspecting six top-level topics shown in the initial topic hierarchy (Fig. 2). In Fig. 2, there are clear and coherent topics like 'ocean, planet, water, earth, sea' (T6: limegreen) and 'patient, disease, cancer, cell, drug' (T7: turquoise). On the other hand, the red topic (T3) with 'girl, love, kid, woman, mother' keywords is more general and ambiguous. The user zooms in to see child topics (Fig. 4). There is a strange topic with keyword 'galleri websit, wix.com, wix' (T19). Upon inspecting its documents, she notices that their contents are actually the same transcript of an advertisement. It turns out that the used web scrapper transcribed youtube commercials instead of the main talk video. She deletes the topic (**RemoveT**). The user continues exploring the unclear red topic by examining its child topics (Fig. 6(left)). She thinks that a red child topic with 'music, song, sing' keywords should be one of the top level topics, so she moves it under the top node (**MoveT**). As a result, there is a top-level topic on 'music, art, artist' in Fig. 6 (middle). The user further splits the blue art topic (**SplitT**). One of its child topics contains both art-related keywords and architecture-related keywords. She moves 'architecture, city, design' keywords to create a sibling topic (**CreateTW**). As a result, the art topic has three sub-topics on art, architecture, and music (Fig. 6 (middle)). Satisfied with the blue topic, she moves on to the brown

[1] `http://www.patentsview.org/download/`, March 12, 2019 Version

[2] Source: `https://github.com/saranyan/TED-Talks`

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2020.2981456, IEEE Transactions on Visualization and Computer Graphics

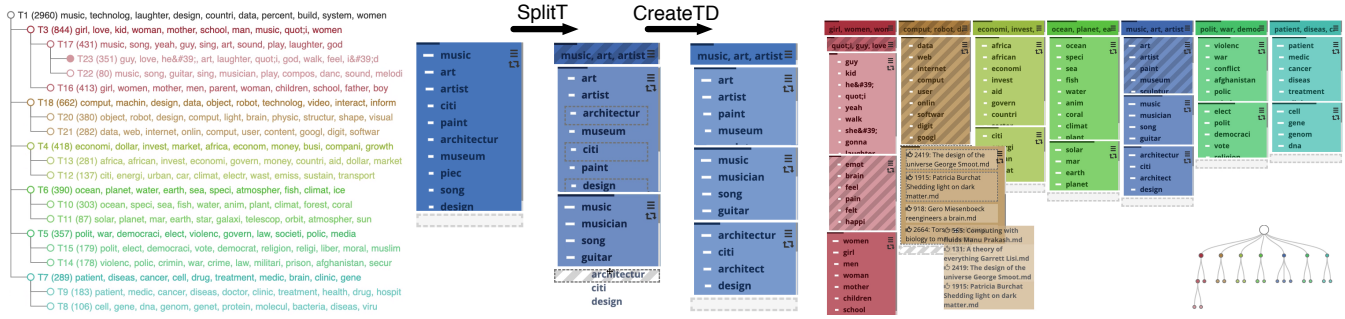


Fig. 6: The topic hierarchy after removing T19 (left). Split a topic into three sub-topics (middle). Create sibling topic with documents (right).

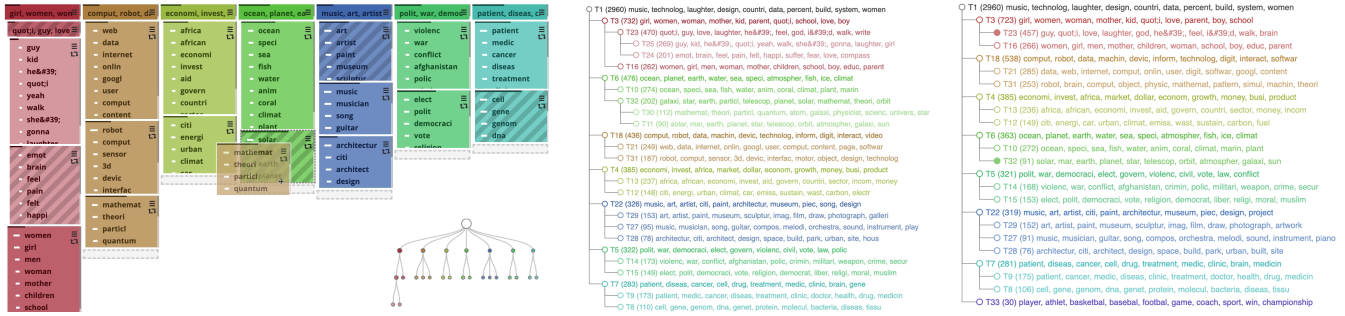


Fig. 7: Merging two topics. Interaction assistant recommends which topic to merge into (left). The topic hierarchy after merging (middle). The final topic hierarchy (right).

topic, which is the second largest. The brown topics are mostly about computer technologies, but she notices that a few keywords 'mathemat, physic, simul, theori' are about natural sciences. She flips the topic and starts to drag several documents about physics and simulation with the goal of separating those out. While dragging, the interaction assistant recommends candidate drop zones with line patterns (Fig. 6 (right)). She decides to move the documents under the top brown topic to create a sibling topic (**CreateTD**). As a result, a new sub-topic about 'mathemat, theori, particl, quantum, galaxi' is created (Fig. 7 (left)). She merges the new topic with another natural science topic on 'solar, mar, earth' (lime green topic in Fig. 7 (left)). She is interested in sports, but she has not seen sports-related topics so far. She types in sports-related keywords such as sports, athletes, basketball, tennis, etc. to create a new topic (**CreateTW**). The new topic has a very small number of documents (30 talks in Fig. 7 (right)). She goes over each document and finds out that some talks are science/tech related (e.g., math behind basketball) or inspirational talks from athletes (e.g. Billie Jean King). She concludes that there are not many TED talks on the topic of sports and finishes the analysis.

6.2 Patent Dataset

The Patent office manages historical patents; and granted or rejected patent applications. Due to technology advances in various fields, there is a need to update the patent classification system. Using ArchiText, we will use interactive topic modeling to make sense of existing utility patents and build a new taxonomy based on their content. The dataset description can be found in Section 5.

A patent officer explores the initial topic hierarchy in Fig. 8. He notices that a brown topic about semiconductor process (red box in Fig. 8 (left)) is grouped with a brown chemistry topic. He moves the semiconductor topic under the top node (**MoveT**). As a result, two topics are separated (red boxes in Fig. 8 (middle)). He inspects one of the chemistry sub-topics and decides to move words (**MoveW**) about pharmaceutical patents (green arrow in Fig. 8 (middle)), which results in a new topic about 'pharmaceut, diseas, treat, treatment' compounds under the chemistry parent topic. Moving on to the rest of the topics, he splits a 'light, image' topic (blue box in Fig. 8 (right)) into an optics topic and an image-related topic (**SplitT**). He delves into sub-topics of all topics, and finds a media related topic under the green 'data, inform' topic (second from the left). Wanting to gather all media related patents under a single top-level topic, he adds keywords 'audio, multimedia, video' into the purple image-related topic (**AddW**) while fixing other topics (**FixT**). As a result, the purple topic becomes larger and contains more media-related patents, some of which are moved from the yellowgreen topic (T13 in Fig. 10). He is now satisfied with the hierarchy and creates the new taxonomy based on the result.

7 Discussion

Interactive topic modeling systems, in order to steer the underlying models, users provide supervision in terms of user interaction. For the same user interaction, there are numerous ways to interpret intermediate results to understand the algorithmic updates depending on the visualization systems and their underlying algorithms. Two basic common factors that can be applied to all interactive topic modeling techniques are the *scope* and the *amount* of user control. First, *scope* determines how wide the impact of the interaction would be. For instance, when a user adds a document to a topic, we can safely assume that the user wants to update the interacted topic. Should this interaction affect only the interacted topic? Or should it also affect the neighboring topics? Or all the topics? Updating in local scope can be faster with less precise results. On the other hand, updating in global scope can provide more accurate results, but may cause unexpected changes in other parts of the model. Next, the *amount* of user control determines to what degree to apply the supervision. For example, when a user adds a keyword to a topic, the user expects the interacted topic to be (more) about the added keyword. In this case, should the updated topic have that keyword as its top keyword at any cost (hard supervision)? Or is increasing the importance weight of that keyword for the topic enough (soft supervision)? What if the topic and its corresponding documents are not related to that keyword, e.g., adding an irrelevant keyword? Some may prefer applying the hard supervision while others may argue for a more truthful representation of the data. In order to balance these trade-offs, we take a simple approach. Our system uses **recompute** operations to supervise the underlying model. **recompute** solves a constrained Nonegative Matrix Factorization (NMF) for the sibling topic nodes (local scope) of the interacted topic with two parameters α and β (Eqn. 2). We considered an option to let the users decide the amount of user control during each interaction, but decided against it. It can be burdensome to the users and it may slow down the analytic process.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2020.2981456, IEEE Transactions on Visualization and Computer Graphics

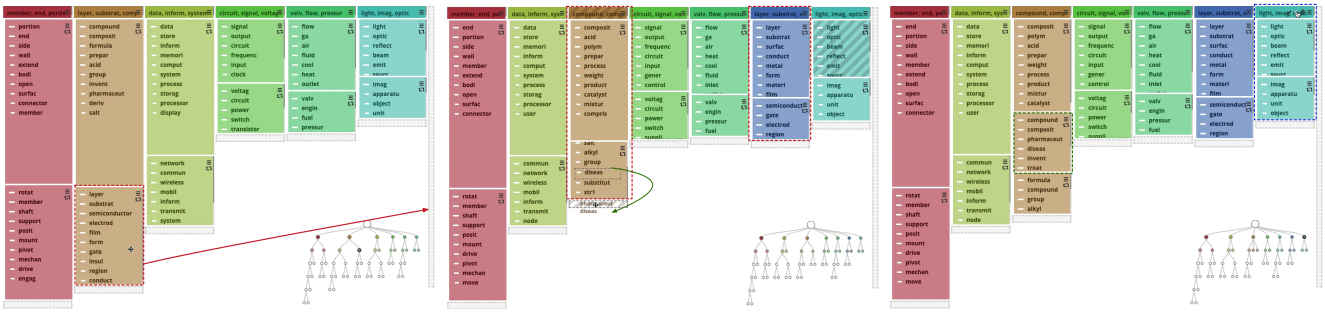


Fig. 8: Initial topic hierarchy of patents (left). After moving a topic (middle). After moving keywords (right).

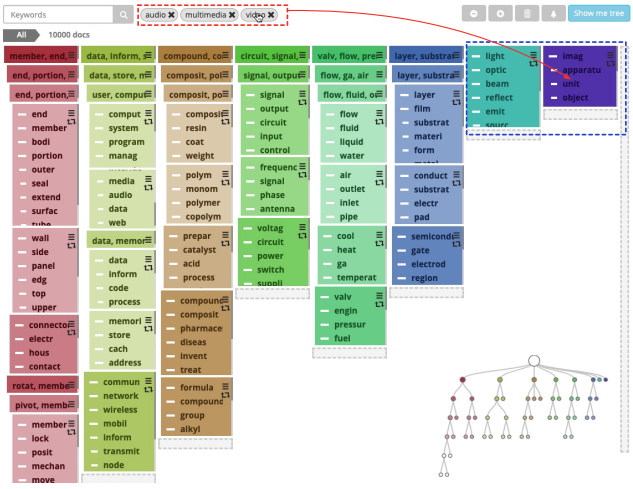


Fig. 9: After splitting a topic into two topics (blue box). Add keywords into the purple topic.

Fig. 10: The final topic hierarchy.

8 Conclusion

In this paper, we proposed interactive hierarchical topic modeling with tight integration among algorithm, visualization, and users interaction. Unlike some previous interactive systems which offer rather limited interaction functionality that may result in unexpected outcomes, our tightly integrated system incorporates user intentions flexibly without strange side effects. In addition, compared to existing interactive topic modeling systems that are not scalable, our system can handle large datasets. As a proof of concept, we developed ArchiText, a prototype system for interactive hierarchical topic modeling and showcased usage scenarios using real-world datasets.

For future work, we plan to take a more proactive approach for smart, convenient human-in-the-loop topic modeling. With tightly integrated topic modeling, users' knowledge and intentions can be flexibly incorporated step by step. In addition to this, we would like our system to remember and learn from previous interactions in order to predict and guide the users' next steps to expedite the model steering process.

Acknowledgments

The authors wish to thank Hyemin Hwang for her help with programming for the visual analytics system. This work was supported in part by the NSF grant OAC-1642410. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of funding agencies.

References

[1] Å. Björck, H. Park, and L. Eldén. Accurate downdating of least squares solutions. *SIAM Journal on Matrix Analysis and Applications*, 15(2):549–568, 1994.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[3] M. Brehmer, S. Ingram, J. Stray, and T. Munzner. Overview: The design, adoption, and analysis of a visual document mining tool for investigative journalists. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2271–2280, Dec 2014.

[4] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 83–92, 2012.

[5] N. Cao and W. Cui. *Introduction to text visualization*. Springer.

[6] N. Cao, D. Gotz, J. Sun, Y.-R. Lin, and H. Qu. Solarmap: Multifaceted visual analytics for topic exploration. In *IEEE International Conference on Data Mining (ICDM)*, pages 101–110. IEEE, 2011.

[7] N. Cao, J. Sun, Y.-R. Lin, D. Gotz, S. Liu, and H. Qu. FacetAtlas: Multifaceted visualization for rich text corpora. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1172–1181, Nov 2010.

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

[9] J. Chuang, Y. Hu, A. Jin, J. D. Wilkerson, D. A. McFarland, C. D. Manning, and J. Heer. Document exploration with topic modeling: Designing interactive visualizations to support effective analysis workflows. In *NIPS workshop on Topic Models: Computation, Application, and Evaluation*, 2013.

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

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

[12] R. Du, D. Kuang, B. Drake, and H. Park. DC-NMF: Nonnegative matrix factorization based on divide-and-conquer for fast clustering and topic modeling. *Journal of Global Optimization*, 68(4):777–798, Aug. 2017.

[13] M. El-Assady, R. Sevastjanova, F. Sperrle, D. Keim, and C. Collins. Progressive learning of topic modeling parameters: A visual analytics framework. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1, 2017.

[14] M. El-Assady, F. Sperrle, O. Deussen, D. A. Keim, and C. Collins. Visual Analytics for Topic Model Optimization based on User-Steerable Speculative Execution. *IEEE Transactions on Visualization and Computer Graphics*, 2018.

[15] A. Endert. *Semantic Interaction for Visual Analytics: Inferring Analytical Reasoning for Model Steering*, volume 4. Morgan & Claypool Publishers, 2016.

[16] A. Endert, P. Fiaux, and C. North. Semantic interaction for visual text analytics. In *Proc. the SIGCHI Conference on Human Factors in Computing Systems*, pages 473–482. ACM, 2012.

[17] A. Endert, C. Han, D. Maiti, L. House, and C. North. Observation-level Interaction with Statistical Models for Visual Analytics. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 121–130, 2011.

[18] N. Gillis, D. Kuang, and H. Park. Hierarchical clustering of hyperspectral images using rank-two nonnegative matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 53(4):2066–2078, 2015.

[19] D. Herr, Q. Han, S. Lohmann, and T. Ertl. Hierarchy-based projection of high-dimensional labeled data to reduce visual clutter. *Computers & Graphics*, 62:28 – 40, 2017.

[20] E. Hetzler and A. Turner. Analysis experiences using information visualization. *IEEE Computer Graphics and Applications*, 24(5):22–26, Sept 2004.

[21] T. Hofmann. Probabilistic latent semantic indexing. In *Proc. the ACM SIGIR conference on Research and Development in Rnformation Retrieval*, pages 50–57. ACM, 1999.

[22] E. Hoque and G. Carenini. ConVisIT: Interactive topic modeling for exploring asynchronous online conversations. In *Proc. the Conference on Intelligent User Interfaces*, pages 169–180. ACM, 2015.

[23] E. Hoque and G. Carenini. Interactive topic hierarchy revision for exploring a collection of online conversations. *Information Visualization*, 2018.

[24] E. Horvitz. Principles of mixed-initiative user interfaces. In *Proc. the SIGCHI Conference on Human Factors in Computing Systems*, pages 159–166. ACM, 1999.

[25] Y. Hu, J. Boyd-Graber, B. Satinoff, and A. Smith. Interactive topic modeling. *Machine Learning*, 95(3):423–469, Jun 2014.

[26] H. Kim, J. Choo, H. Park, and A. Endert. Interaxis: Steering scatterplot axes via observation-level interaction. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):131–140, 2016.

[27] J. Kim and H. Park. Fast nonnegative matrix factorization: An active-set-like method and comparisons. *SIAM Journal of Scientific Computing*, 33(6):3261–3281, Nov. 2011.

[28] M. Kim, K. Kang, D. Park, J. Choo, and N. Elmqvist. TopicLens: Efficient multi-level visual topic exploration of large-scale document collections. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):151–160, Jan 2017.

[29] D. Kuang and H. Park. Fast rank-2 nonnegative matrix factorization for hierarchical document clustering. In *Proc. the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 739–747. ACM, 2013.

[30] B. C. Kwon, H. Kim, E. Wall, J. Choo, H. Park, and A. Endert. Axisketcher: Interactive nonlinear axis mapping of visualizations through user drawings. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):221–230, 2017.

[31] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proc. the Neural Information Processing Systems*, pages 556–562, 2000.

[32] H. Lee, J. Kihm, J. Choo, J. Stasko, and H. Park. iVisClustering: An interactive visual document clustering via topic modeling. *Computer Graphics Forum*, 31(3pt3):1155–1164, 2012.

[33] 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. *International Journal of Human-Computer Studies*, 105:28 – 42, 2017.

[34] Y.-R. Lin, J. Sun, N. Cao, and S. Liu. Contextour: Contextual contour visual analysis on dynamic multi-relational clustering. In *Proc. the SIAM International Conference on Data Mining*, pages 418–429. SIAM, 2010.

[35] S. Liu, J. Yin, X. Wang, W. Cui, K. Cao, and J. Pei. Online visual analytics of text streams. *IEEE Transactions on Visualization and Computer Graphics*, 22(11):2451–2466, Nov 2016.

[36] N. E. Miller, P. C. Wong, M. Brewster, and H. Foote. Topic islands-a wavelet-based text visualization system. In *Proc. Visualization*, pages 189–196, Oct 1998.

[37] C. Musialek. Distill-ery: Iterative topic modeling to improve the content analysis process.

[38] T. Mhlbacher, H. Piringer, S. Gratzl, M. Sedlmair, and M. Streit. Opening the black box: Strategies for increased user involvement in existing algorithm implementations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1643–1652, Dec 2014.

[39] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin. Automatic evaluation of topic coherence. In *Proc. the Human Language Technologies: NAACL-HLT*, pages 100–108. Association for Computational Linguistics, 2010.

[40] H. Park and L. Eldn. Downdating the rank-revealing urv decomposition. *SIAM Journal on Matrix Analysis and Applications*, 16(1):138–155, 1995.

[41] F. V. Paulovich and R. Minghim. HiPP: A novel hierarchical point placement strategy and its application to the exploration of document collections. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1229–1236, Nov 2008.

[42] W. A. Pike, J. Stasko, R. Chang, and T. A. O'connell. The science of interaction. *Information Visualization*, 8(4):263–274, 2009.

[43] A. Smith, T. Hawes, and M. Myers. Hiearchie: Visualization for hierarchical topic models. In *Proc. the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 71–78, 2014.

[44] A. Smith, V. Kumar, J. Boyd-Graber, K. Seppi, and L. Findlater. Closing the loop: User-centered design and evaluation of a human-in-the-loop topic modeling system. In *Proc. the International Conference on Intelligent User Interfaces*, pages 293–304, New York, NY, USA, 2018. ACM.

[45] A. Smith, T. Y. Lee, F. Poursabzi-Sangdeh, J. Boyd-Graber, N. Elmqvist, and L. Findlater. Evaluating visual representations for topic understanding and their effects on manually generated topic labels. *Transactions of the Association for Computational Linguistics*, 5:1–16, 2017.

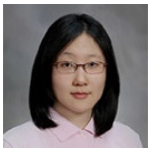
[46] J. Stasko and E. Zhang. Focus+ context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *IEEE Symposium on Information Visualization*, pages 57–65. IEEE, 2000.

[47] E. Wall, S. Das, R. Chawla, B. Kalidindi, E. T. Brown, and A. Endert. Podium: Ranking data using mixed-initiative visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):288–297, 2018.

[48] X. Wang, S. Liu, J. Liu, J. Chen, J. Zhu, and B. Guo. Topicpanorama: A full picture of relevant topics. *IEEE Transactions on Visualization and Computer Graphics*, 22(12):2508–2521, Dec 2016.

[49] Y. Yang, Q. Yao, and H. Qu. Vistopic: A visual analytics system for making sense of large document collections using hierarchical topic modeling. *Visual Informatics*, 1(1):40 – 47, 2017.

[50] K. Yoo and H. Park. Accurate downdating of a modified gram-schmidt qr decomposition. *BIT Numerical Mathematics*, 36(1):166–181, Mar 1996.



Hannah Kim is currently a Ph.D. student in computer science at Georgia Institute of Technology. Her research interests include data mining, machine learning, and visual analytics.

Alex Endert is an Assistant Professor in the School of Interactive Computing at Georgia Tech. He directs the Visual Analytics Lab, where he leads research that explores novel user interaction techniques for visual analytics. His lab often applies these fundamental advances to domains including intelligence analysis, cyber security, manufacturing, and others.

Barry L. Drake is a senior research faculty member at Georgia Tech in the Information and Communications Laboratory (ICL) of the Georgia Tech Research Institute (GTRI) and in the School of Computational Science and Engineering. His research interests include adaptive algorithms, learning machines, numerical linear algebra, and applying these technologies to solve real-world problems.

Haesun Park is a professor in the School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, Georgia, U.S.A. She was elected as a SIAM Fellow in 2013 and IEEE Fellow in 2017. Her research interests include numerical computing, large-scale data analysis, visual analytics, text mining, and parallel computing.