DOI: 10.1111/cgf.14301
Eurographics Conference on Visualization (EuroVis) 2021
R. Borgo, G. E. Marai, and T. von Landesberger
(Guest Editors)

Volume 40 (2021), Number 3

Learning Contextualized User Preferences for Co-Adaptive Guidance in Mixed-Initiative Topic Model Refinement

F. Sperrle, H. Schäfer, D. Keim, and M. El-Assady

University of Konstanz

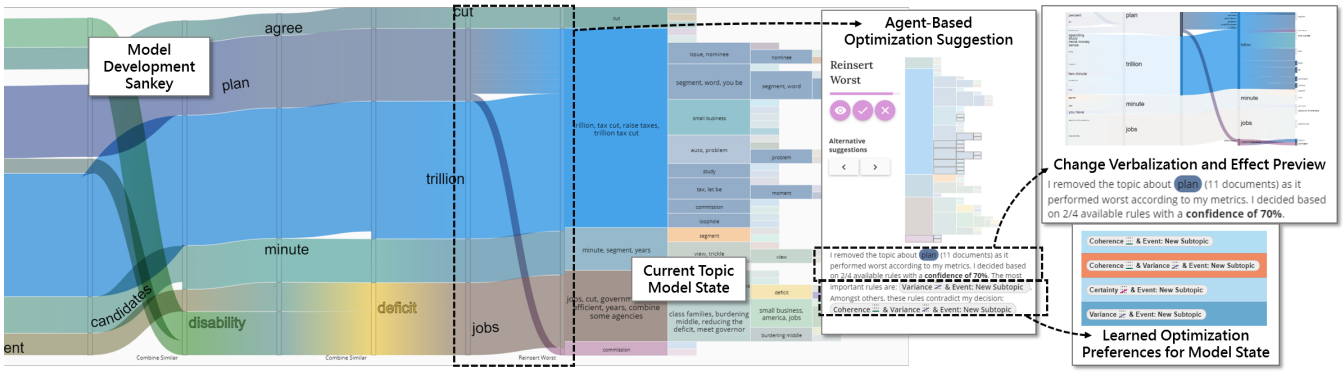


Figure 1: *In a co-adaptive guidance process, optimization agents suggest refinement operations that users can accept or reject. From this feedback, agents learn a contextualized preference model that encodes in which analysis states their guidance is meaningful to the user. Each agent provides a preview of its suggested model states and summarizes its suggested changes together with a justification.*

Abstract

Mixed-initiative visual analytics systems support collaborative human-machine decision-making processes. However, many multi-objective optimization tasks, such as topic model refinement, are highly subjective and context-dependent. Hence, systems need to adapt their optimization suggestions throughout the interactive refinement process to provide efficient guidance. To tackle this challenge, we present a technique for learning context-dependent user preferences and demonstrate its applicability to topic model refinement. We deploy agents with distinct associated optimization strategies that compete for the user's acceptance of their suggestions. To decide when to provide guidance, each agent maintains an intelligible, rule-based classifier over context vectorizations that captures the development of quality metrics between distinct analysis states. By observing implicit and explicit user feedback, agents learn in which contexts to provide their specific guidance operation. An agent in topic model refinement might, for example, learn to react to declining model coherence by suggesting to split a topic. Our results confirm that the rules learned by agents capture contextual user preferences. Further, we show that the learned rules are transferable between similar datasets, avoiding common cold-start problems and enabling a continuous refinement of agents across corpora.

1. Introduction

Mixed-initiative approaches in which human and computer agents contribute their best-suited actions at the most appropriate time [AGH99] have proven useful for the optimization of machine learning models. Due to the complex nature of many machine learning tasks, such refinements usually operate on multiple optimization objectives simultaneously. These objectives are *proxy-measures* for the quality of the trained machine learning model but often do not capture the user's intuition of quality. Additionally, there often exist multiple, equally correct solutions that fulfill different user preference profiles. In particular, the *desired* outcome of complex analysis tasks in various domains like text analysis or crime investigation often depends on the analyst's personal preference, the domain itself, or any downstream tasks that require certain results. A major concern of domain experts in these fields is to incorporate their nuanced domain understanding to adapt the machine learning results to their preferences, which, in turn, depend on the context and task at hand. This makes those tasks prime targets for visual analysis, where domain experts can incorporate their knowledge.

To enable more efficient mixed-initiative processes in these circumstances, visual analytics approaches should allow for co-adaptation between users and systems [SJB*20]. In this paper, we

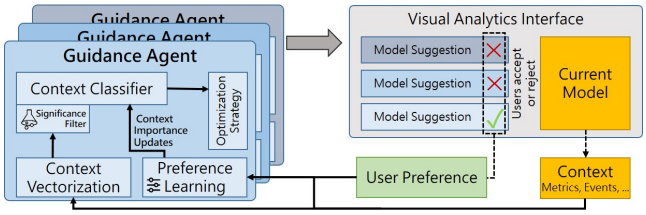


Figure 2: *Agents make suggestions according to a fixed optimization strategy. By observing direct and indirect user feedback, they refine a classifier used to decide in which contexts to provide guidance.*

focus on interactive topic model refinement and present a guidance technique that relies on *contextualized, adaptive agents* and is conceptualized in Figure 2. Each of the task-specific agents is responsible for a single optimization operation in the refinement process. During the model refinement, all agents compete for the user's satisfaction by suggesting alternative model states based on the current analysis context. Based on direct and indirect user feedback, each agent aims to capture the users' preferences by learning in which analysis contexts to provide its suggestions. Thus, each agent strives to deliver only suggestions that would be accepted by the user while not making suggestions that would be rejected. This ensures that the visual analytics process can progress with *minimum feedback for maximum gain* [EASS*18] by reducing the number of unwanted suggestions interrupting the user's analysis process.

We instantiate the proposed technique in an application for the iterative refinement of the *Incremental Hierarchical Topic Model* [EASD*19] (IHTM). The application uses guidance agents that suggest common topic model refinement operations and explain and justify their suggestions using verbalizations. Ignored suggestions decay over time, reducing the required amount of active system teaching. We evaluate our technique using different approaches. First, a qualitative expert user study to assess the usability of the interactive interface. Second, three quantitative experiments to verify the learning ability of our technique for different aspects: preference-based topic ranking, contextualization, as well as transferability. Our multi-faceted evaluation confirmed the effectiveness of our approach for mixed-initiative model refinement and suggests that agents learn meaningful optimization rules that match user intuition but could be improved through the addition of domain semantics.

To summarize, the main contributions of this paper are: (1) A technique for providing agent-based, co-adaptive guidance for multi-objective optimization problems in visual analytics. (2) An application for the guided refinement of topic models that instantiates this technique. (3) A multi-faceted evaluation to assess the interplay of different aspects influencing the effectiveness of adaptive guidance and illustrates the validity of the learned user preference models.

2. Background and Related Work

Recent years have seen a rise of interest in guidance in visual analytics. Ceneda et al. characterized guidance processes in terms of a knowledge gap encountered by the user, the available inputs and outputs, and the degree of guidance [CGM*17]. More recently, they also characterize guidance as an adaptive mixed-initiative process [CGM19]. In our previous work, we extended that characterization to define *co-adaptive guidance* as a mixed-initiative interaction process characterized by the dynamics of learning and teaching [SJB*20]. In this paper, we present a technique that employs the concept of co-adaptive guidance to enable users to teach a set of guidance agents their personal preferences. Collins et al. [CAS*18] have provided a process-oriented model of guidance and state "just-in-time-visualization" to be an important goal of guidance.

Guidance and Recommendation Interfaces – Both Ceneda et al. [CGM19] and Collins et al. [CAS*18] survey a large number of applications that employ guidance. In particular, Ceneda et al. [CGM19] identified twelve papers that use guidance in the model building process. Several systems provide guidance for splitting data into clusters or decision trees [AEK00, AAR*09]. Drucker et al. support iterative document clustering by highlighting relevant clusters once users select an artifact [DFB11]. While these approaches provide guidance upon request, we propose a technique in which agents learn to provide guidance automatically when appropriate. This paper was loosely inspired by cooperative contextual bandits [TvdS15], a class of recommender systems that have recently been used successfully in personalized design ideation for the creation of mood boards [KLHO19]. However, this approach relies on partitioning the design space of mood boards according to multiple dimensions and assigning agents to each partition that then sample suggestions from their partition. This is not possible for the solution space of topic models or text-based algorithms in general. Instead, our technique uses guidance agents that each focus on presenting guidance according to a specific, fixed optimization strategy.

Document Exploration and Topic Model Optimization – A large number of visual analytics applications for topic model optimization have been proposed. Some low-level approaches operate directly on the level of keyword constraints [CLRP13, HBGSS14]. More high-level approaches enable the user to select one of multiple provided refinement operations [CLRP13, HC15, DYW*13]. These operations inspired the refinement operations offered by our application. However, the respective approaches do not provide any active guidance in the modeling process. Recently, Kim et al. [KDEP20] presented ArchiText, a scalable approach to iterative topic modeling that tightly integrates a large set of operations proposed in previous work with a bespoke user interface. While this system enables iterative refinement, it only has minimal guidance capabilities.

Rather than working with topics directly, Park et al. [PKL*18] present a system in which users iteratively refine a set of concepts that capture semantics and can be used to describe documents. *Semantic Concept Spaces* [EAKC*20] bridges the gap to topic modeling by allowing users to refine perceived similarities between concepts and using the learned similarity information to adjust a topic model. The system uses semantic interactions [EFN12] to suggest additional concept space changes to the user based on the adapted similarity relations, thus indirectly guiding them in refining the topic model. In contrast to this approach of learning the user's perceived concept similarity, our approach is content-agnostic. It operates on a vector space defined by topic model quality metrics, where it learns which optimizations are preferred by users in which contexts.

The work closest to our approach is that of Speculative Execution for Topic Model Optimization [EASD*19]. Our approach builds on the same framework, using an iteratively-created model and the

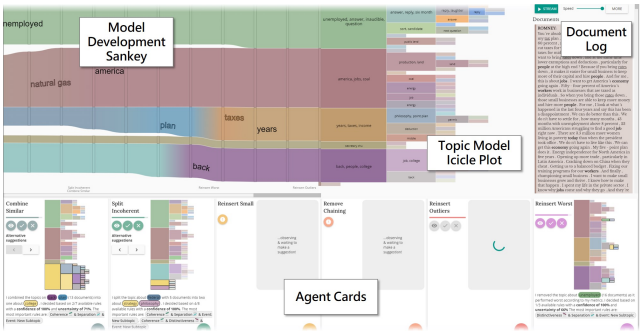


Figure 3: *Overview of how the four main system components are arranged in the user interface to instantiate our technique.*

same refinement operations. In contrast to this work, which focused on explaining the model's decision-making processes, we present a technique for learning and generating adaptive, contextualized guidance, which was identified as a limitation in the previous study.

3. Guided Topic Model Refinement

Among others, scholars of the humanities use topic modeling as a means to gaining an understanding of text corpora before using the obtained results as a basis for downstream research efforts. Optimizing models to align with their understanding incorporates several tasks like understanding and diagnosing models, re-assigning documents, and refining topic hierarchies [EASD*19, EAKC*20]. Often, this presents a challenge due to a lack of technical knowledge and the amounts of analyzed data. Co-adaptive guidance in which system and user simultaneously learn from each other and teach one another [SJB*20] can make this process more efficient: systems teach the user model optimizations, and users teach the system which types of suggestions are useful in which contexts. To that end, we target three design goals concerning the guidance process:

G1: Enable contextualized preference learning. To refine in which contexts guidance should be provided, the system should gather implicit and explicit user feedback to suggestions and learn which suggestions are relevant in which contexts. As the user's time is the limiting factor in this learning process, the system should maximize what it can learn from a minimal amount of interactions.
G2: Provide contextualized suggestions tailored to the current analysis state. To achieve this goal, we employ guidance agents that are responsible for a distinct action and aim to only take the initiative in situations where their suggestion would be meaningful.
G3: Users must remain in control of the refinement process. As mixed-initiative interfaces aim to combine the strengths of humans and machines, they strike a balance between manual labor and automation. However, while systems might aim to teach users relevant optimizations, users should remain in control of the process. This also implies that the system's guiding suggestions should not interfere with the ongoing refinement that users are performing.

To achieve these goals and enable effective guidance, we want to avoid overwhelming users with a complex, unintuitive interface. Thus, we follow these design rationales and interaction principles:

I1: Simple to use. The system should avoid complex interaction patterns or model visualizations. To assess the current model state, we provide an initial overview first and provide access to both details and the concrete documents in a topic on demand. To flatten the learning curve of the system, we employ consistent visual representations of all major entities (topics, documents, keywords) and make the same interactions available on all occurrences of an entity. Further, we connect all views using linking and brushing and use similar colors for semantically similar topics.
I2: Intelligible and Trustworthy. Each agent enables a detailed inspection of its suggestions and provides a verbal summary and explanation. In uncertain cases, this explanation also reveals evidence considered by agents that would contradict making a suggestion. By including those factors, we aim to foster trust in users that the agents consider a holistic view of the analysis process. Throughout the system, uncertainty is visually encoded through opacity.
I3: Focus on Guided Refinement. The guidance suggestions provided by the agents take a prominent role in the refinement process, and a large part of the interface is allocated to their presentation. Nonetheless, manual refinement of topics with drag and drop is still possible to resolve situations not readily addressed by the agents.
I4: Provide Model History. A model history is particularly important in mixed-initiative processes as it can highlight the collaborative nature of the refinement and allows users to revisit previous agent suggestions at a later point in time.

4. Topic Model Refinement Interface

Before we present our technique for contextualized guidance generation in section 6 we introduce our system that instantiates this technique in the following two sections. It is depicted in Figure 3 and consists of four main views: the *topic icicle* shows the current modeling state, a *development Sankey* highlights model changes over time, the *document log* enables close reading, and six *agent cards* present the agents' guidance suggestions. The system enables users to observe the incremental nature of the model building process: over time, documents are first added to the document log before updating the topic icicle and triggering a new model state captured in the Sankey.

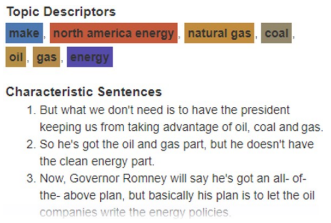
4.1. IHTM: Incremental Hierarchical Topic Model

The implemented system is built around *IHTM*, an **I**ncremental, **H**ierarchical **T**opic **M**odel [EASD*19]. In contrast to other models like LDA [BNJ03], IHTM is deterministic and avoids frequent issues with probabilistic models where users cannot replicate their previously obtained results. In addition to being deterministic, IHTM is an incremental model that builds a hierarchical topic structure by sequentially adding new documents to a tree structure. Leaf nodes in IHTM's hierarchical topic structure represent documents, and inner nodes represent (sub)topics. To insert a new document, the algorithm searches the most similar tree node through recursive, breadth-first descend. Documents are represented as tf-idf vectors and tree nodes as the average of all contained document vectors. All vectors are compared using cosine similarity. If the recursive descend finds an inner node of the tree (representing a (sub)topic), the document is added as an additional child. If the found node is a leaf node, a new subtopic containing the existing leaf and the new document will be created. The iterative model building process allows for computationally inexpensive incremental model building. We argue that this constitutes an advantage over non-incremental black box models like LDA for two reasons. First, users can follow the model building

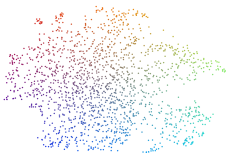
process and identify potential modeling problems early. Second, fixing modeling issues early in the process directly influences the created topic tree, removing the need for multiple modeling iterations.

4.2. Model Visualizations

The topic tree produced by the IHTM is visualized (and continuously updated) as a rotated **icicle plot** in which the hierarchical topics are depicted from left to right. Each node's width corresponds to its level in the hierarchy, and higher-level topics take up more space. The rightmost nodes in each cluster represent the documents contained in the respective (sub)topics. Users can directly incorporate their domain knowledge into the modeling result by dragging and dropping subtopics from the icicle plot to make isolated changes (**I3**). To visually distinguish document nodes from topics, they are shorter and lighter in color. All topic rectangles contain the most important topic descriptors keywords. The number of descriptors is scaled linearly with the topic level to show more information about more influential higher-level topics. A **tooltip** for each topic reveals additional details (**I1**) and shows all of its extracted descriptor keywords and their associated colors (see below) and up to ten representative sentences. Tooltips for document nodes show the document's text. Clicking on a topic node opens a **popup window** that re-iterates the topic descriptors and representative sentences and adds a text view of all documents contained in the topic. Furthermore, users can zoom the icicle by control-clicking on topics, enabling the system to scale to large models.



To generate the **colors** for all topics and documents, we adapt the idea of *Semantic Concept Spaces* [EAKC*20] in which similar keywords are mapped to similar colors in a 2D color map. We first enumerate all keywords from the corpus and remove stop words. Additionally, the system allows the removal of user-defined, context-specific stop words. For the remaining keywords, we obtain DistilBERT [SDCW20] embedding vectors that we project to 2D using UMAP [MHM18]. We then map the resulting projection space to a 2D color map [BLBS11] and assign each keyword a color. To obtain the color for a topic, we calculate the average DistilBERT embedding of its keyword descriptors, weighting each descriptor by its relevance score produced by the keywords extraction algorithm YAKE [CMP*20]. We again project this average embedding using UMAP and assign the topic the associated color. Removing keywords contained in the aforementioned keyword blacklist has two benefits. First, it ensures that keywords with more descriptive content can be projected with less distortion. Second, it keeps these keywords from taking up room in the color map and leaves more colors available for other keywords and concepts.



To show the temporal development of top-level topics (**I4**), we employ the **model development Sankey** diagram shown in Figure 1. We place the Sankey adjacent to the icicle, and each node represents the topic directly next to it in the icicle. Each new document inserted into the model adds a column to the right of the Sankey diagram. Older model states are pushed out of view but can be reached by scrolling the visualization. The Sankey links show how many documents are shared between two (states of) topics and can be used to track how both manual refinements and agent suggestions moved documents between topics. In addition to showing document movement, the Sankey also shows the development of topic descriptors. Whenever the top-ranked descriptor of a topic changes, the new descriptor is displayed at the corresponding model state's position in black. The font size is scaled to the number of documents in the topic. Changing descriptors typically lead to a change in topic color. We display outdated descriptors in their topic's old color, making them stand out less. Together with descriptors only being replaced when they change rather than at fixed intervals, this shifts user focus to currently developing topics. The same topic interactions (hovering for a tooltip, clicking for a popup) that users are familiar with from the icicle plot are also available in the Sankey.

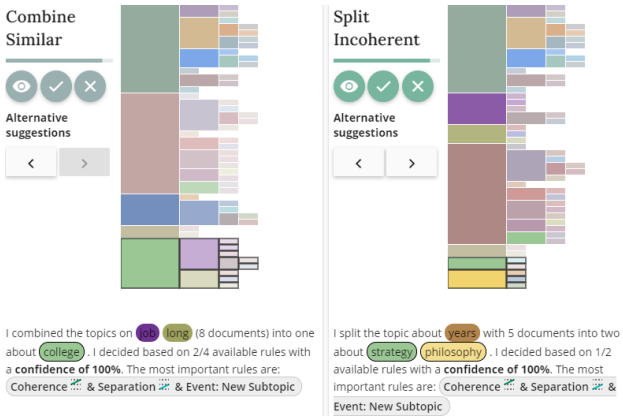


Figure 4: *Agents preview their suggested changes in a small icicle and verbalize them. They also explain their classification and state their confidence and certainty.*

The **document log** shows all documents inserted into the model. New documents are placed on top of the list; older ones remain accessible by scrolling. The most important keywords in each document are highlighted to aid in quickly scanning a document. All documents are colored using the same process as topics in the icicle plot.

5. Agent-Based Guidance Interface

The system provides six different guidance agents that perform distinct optimizations like splitting or merging topics. All agents observe the modeling process and provide suggestions only when they expect them to be helpful. To avoid interrupting users, the suggestions do not stop the stream of documents from being inserted into the model. Instead, agents automatically keep their suggestions updated to reflect newly added documents. All available agents and the underlying technique for learning contextualized user preferences will be introduced in section 6. This section focuses on the interaction possibilities that the guidance agents afford (**I3**). We decided against representing each agent as a virtual avatar and instead show them as **agent cards** in the interface for simplicity and clarity. Two examples of agents providing suggestions are shown in Figure 4. Figure 3 also shows agents without current suggestion.

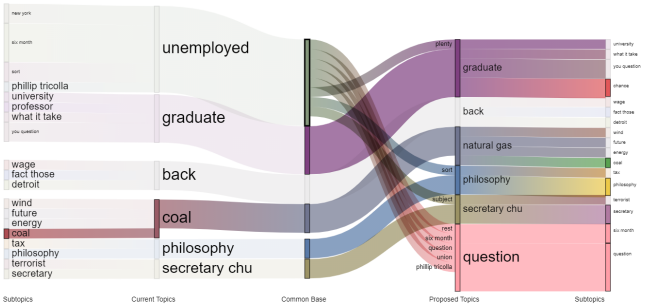


Figure 5: *The model diff Sankey shows changes between the current and a suggested model with respect to a common ancestor model.*

Previous work has successfully applied verbalization to explain machine learning models [HSD19] and presented a design space for verbalization [SBE*18]. We employ template-based *overview and detail* verbalization [SBE*18] to provide a summary of agent suggestions at the bottom of the card. The summary (see Figure 4) contains information on existing and new topics, as well as the number of affected documents (**I2**). Additional information on those topics is available through the tooltip and popup introduced in the previous section (**I1**). In addition to the summary, each agent also states its *confidence* in the suggestion based on its internal classifier (**I2**) and adds a warning when its *uncertainty* goes above 50%.

The **model preview icicle** on the right-hand-side of the agent card shows how the model would look if the suggestion were accepted; the changed topics mentioned in the summary and their documents are highlighted with a grey border. The preview's opacity is determined by the agent's uncertainty, with more uncertain suggestions having a higher opacity and visually fading in comparison to other suggestions. More detailed information on how the model would change and which specific documents are affected is visualized in the **model diff Sankey** (see Figure 5) that can be accessed in a popup view via the review button . Since agent suggestions do not stop the iterative modeling process to avoid interrupting users, new documents iteratively added to the model are also added to each suggestion, leading to potentially large differences between model states. The model diff Sankey reveals those differences by matching both model states via a common ancestor state: The first two columns show the subtopics and topics of the current model, respectively. Columns four (topics) and five (subtopics) provide the same information for the model suggested by the agent. Links between the columns indicate to which topic a subtopic belongs. The middle column shows the common ancestors between both models: the current model at the time that the suggestion was initially created. Any topics that changed due to the agent's suggestion are marked with a black border. Links originating from such topics indicate the result of the guidance suggestion and are highlighted. A tooltip reveals which documents were affected by the change. All links connecting topics that are identical between the current model and the suggestion are drawn with low opacity, enabling users to focus on changes between the two model states and to ensure that the agent's suggestion had no unexpected side effects.

Once users have reviewed a suggestion, they can accept or reject it. The following section describes how agents learn from these interactions to capture user preferences. Agents can provide multiple suggestions for different topics in a given situation. Those alternative suggestions, if available, can be cycled through using the previous ‹ and next › buttons. Due to the iterative nature of the process, suggestions slowly become outdated. This effect is visualized by a continuously decreasing progress bar in each agent's card. Suggestions are considered outdated and automatically rejected if they have not been accepted after ten document inserts. When an agent is not currently suggesting an optimization, users can **manually request guidance** , teaching agents that their suggestions might be relevant in the given situation.

6. Adaptive Guidance Generation

To generate adaptive guidance suggestions, we introduce a flexible technique for learning contextualized user preferences. This technique is not specific to topic modeling and generalizes to any metric-based model optimization task. Existing guidance systems have often focused on content-based guidance [SSKEA19] that presents users with suggestions that are semantically similar to their previous interactions. Our technique is content-agnostic and does not rely on knowing specific (semantic) interaction sequences. Instead, it builds on the temporal changes observed in metrics that describe the model.

6.1. Contextualized Guidance Agents

At the core of our technique, we employ **guidance agents** that each target one **distinct optimization strategy**. Over time, agents build a **user preference model** by having their suggestions accepted or rejected in different situations. Six complementary agents support various refinement operations: the *Combine Similar Topics* agent identifies and merges the most similar topics. In contrast, the *Split* agent splits a topic into two new topics. The *Remove Topic Chains* agent removes so-called topic chains, modeling artifacts produced by IHTM in certain conditions [EASD*19]. The remaining three agents, *Reinsert Small Topics*, *Reinsert Outliers*, and *Reinsert Worst Topics*, each identify a certain subset of topics to be removed from the tree and reinserted at a new position using the original IHTM algorithm. Our previous work has introduced all optimizations in more detail and shows that they provide meaningful model alternatives [EASD*19].

As each agent's refinement operation is fixed, its task in the guidance process is simplified from having to provide the right guidance at the right time to providing its associated optimization in the right *model analysis contexts*. To that end, each agent builds and maintains a binary *context classifier*. As a result, our implementation trains six independent classifiers in parallel and allows all agents to make suggestions independent of the other agents. As there is (theoretically) no limit to the number of agents that can be used, limiting each agent to performing only one operation does not manifest as a limitation of the technique. Instead, it enables the use of simpler classification models that can be used to explain each agent's decisions.

6.2. Analysis Context Vectorization

Each agent builds a context classifier over the **model analysis context** to identify in which situations to provide guidance. We define the model analysis context as a vector containing *quality metric measurements* and *modeling events* for a given model state.

In our implementation, we use the five established topic quality metrics *coherence*, *separation*, *variance*, *certainty*, and *distinctiveness* [EASD*19]. A sixth dimension captures the IHTM events *new topic*, *new subtopic*, and *document added* that can influence the quality metrics.

To learn satisfactory classifiers on this six-dimensional model analysis context, agents would require large amounts of training and interaction data. Even if enough training data could be obtained for offline pretraining, the classifiers would likely not be able to adapt to the users' preferences during the analysis session. To tackle this issue, we significantly reduce the space by transforming the *model analysis context* into vectors of metric development shapes that we call **context vectors**. Rather than individual metric values, the context vector contains each metric's development between two points in time. We use seven linear shapes to vectorize the metric development over time with a window size of 1: *low flat*, *medium flat*, *high flat*, *low raise*, *medium raise*, *high lower*, and *medium lower*. The categories *low*, *medium*, and *high* are determined by the first, second, and third tertile, respectively. The direction of change is determined by comparison with the metric value at the previous time step. Metrics with a change of less than one percent are considered to be *flat*. The window size of 1 means that a new vector is created after each document insert. Different instantiations of the technique can freely choose window sizes to consider smaller or larger contexts. We also note that the shapes or shapelets that are useful in a context vector depend on the analysis context. While we rely on seven linear shapes, our proposed technique is generic and works with any number of arbitrary shapes or shapelets.

Although the simplification of the analysis context to context vectors reduces the state space drastically, the number of possible context vectors remains large. To avoid overfitting on a few specific context vectors and not being able to provide meaningful guidance in the majority of states, agents also use **partial context vectors** for the classification. Each n-dimensional context vector $S =< s_1, \ldots, s_n >$ has $2^n - 1$ partial context vectors defined by the power set $\mathcal{P}(S) \setminus S$ of the elements contained in the context vectors. By not considering one or multiple dimensions s_x, these partial context vectors are broader and more general, allowing agents to improve their user preference model to meaningfully cover the *model analysis context* with significantly fewer visited states. Furthermore, avoiding the overfitting of user preference models to specific datasets enables continuous agent learning and allows users to refine their agents' preference models over many visual analytics sessions instead of starting from scratch every time. Our evaluation showcases the transferability of preference models across datasets.

6.3. Context Classification

To determine suitable contexts for providing guidance, each agent builds a classifier that accepts (partial) context vectors and returns a binary decision on whether to provide a suggestion or not. The used classifier should enable continuous refinement and be intelligible to foster trust in an agent's suggestions. We chose a rule-based classifier that we introduce in the following.

As partial context vectors are based on the power set of analysis context dimensions, longer partial context vectors frequently comprise the information encoded in shorter partial context vectors. To avoid over-emphasizing this duplicated information during classification, we adapt the concept of "closed patterns" from sequential pattern mining [ZH02]: the support of a pattern is defined as its number of occurrences, and a pattern is considered *closed* if it is not included in a longer pattern with the same support. The agents store the support of each partial context vector and only use **closed context vectors** during classification. Further, they store how often suggestions provided in a given context c_i were accepted (a_i) or rejected (r_i). When a new model analysis context is encountered during the optimization process, the support of all associated partial context vectors is incremented. As a result, the closedness of a partial context vector can vary over time.

After each document insert into the topic model, each agent performs the steps connected with orange arrows in Figure 6: It vectorizes the analysis context, filters out non-closed contexts, and makes a guidance trigger decision based on all closed context vectors $\mathcal{C}$ using a **linear additive model**. For a context vector $c_i \in \mathcal{C}$, let $w_i = \frac{a_i}{a_i + r_i} - 0.5$ its *weight*. Weights are in $[-0.5, 0.5]$, and larger absolute weights show that suggestions in the respective contexts have received more consistent user feedback. Longer, more specific context vectors are expected to have a higher weight as they appear less frequently and are typically less controversial. To keep the classifier from overfitting on those contexts, we log-normalize all weights by the support of the context, leading to lesser weight differences between short and long partial contexts. While the agents are learning, a significant number of contexts will have low weights close to zero, indicating uncertainty. As those contexts can distort the classification result, we exclude all contexts with an absolute weight below 0.25 from the classifier. This boundary and the normalization parameters introduced above have been chosen experimentally and limit the impact of very frequent contexts. They might require fine-tuning for different instantiations of the technique.

The decision on whether to provide guidance for a filtered and normalized set of context vectors $\mathcal{C}_>$ is determined by summing up the weights of all context vectors: $g_{\mathcal{C}_>} = \sum_{c_i \in \mathcal{C}} g_i$. Agents will make a suggestion if $g_{\mathcal{C}_>} > 0$. The confidence of this decision is defined as the ratio of all scores to all absolute scores: $c = \frac{|\sum_{c_i \in \mathcal{C}_>} g_i|}{\sum_{c_i \in \mathcal{C}_>} |g_i|}$. The uncertainty is given by the number of ignored contexts below the weight threshold: $u = 1 - \frac{||\mathcal{C}_>||}{||\mathcal{C}||}$. Both confidence and uncertainty are displayed in the system to justify each agent's decisions.

To avoid a cold-start of the classification model at the beginning of the first analysis session, we generate initial contexts for each agent. We compiled three subsets of the 20news corpus [Lan], each containing 15 documents each from eight newsgroups for a total of 360 documents. For each of the subsets, we build an IHTM and request a suggestion from all agents after each document insert. We then automatically evaluate the quality of both the suggestions and the original models using *normalized mutual information* [MGH13] (NMI) and extract the context vectors based on the original model. Depending on whether the suggestion improved or worsened the NMI score, we initialize the acceptance or rejection scores of the context vectors to 1, respectively. If an analysis context is encountered multiple times, the values for support, acceptance, and rejection are increased for the respective existing context vectors in the user preference model.

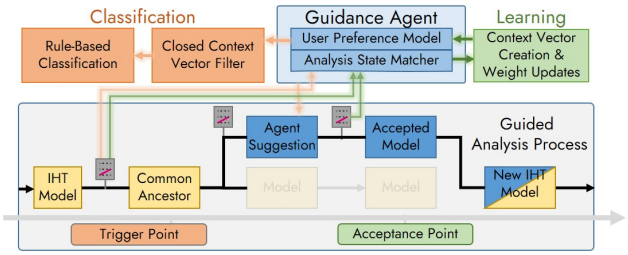


Figure 6: *Agents match the analysis context against their user preference model and classify whether to provide a suggestion. User acceptance (or rejection, not shown) leads to agent learning and replaces the current model with the suggestion.*

6.4. Learning from User Interaction

Agents learn from both implicit and explicit relevance feedback by adapting either their acceptance or rejection score (a_i or r_i) for a given context c_i, in turn influencing all future classification results. This section presents how much the scores of which contexts are updated depending on the feedback type. As the closedness of a partial context vector can change over time, agents do not filter by this property in the learning phase and pre-emptively update the scores for all relevant partial contexts.

In our implementation for topic model refinement, each agent remembers which suggestions it has made before and does not suggest its operation on the same topic(s) more than twice. To avoid decaying to a state where agents cannot make suggestions anymore, this memory is wiped whenever a suggestion from the agent is accepted, or no valid suggestion candidates can be found. In addition, users can always manually request guidance from an agent, teaching them that guidance might be needed in a given context.

Types of Feedback – We distinguish between explicit and implicit feedback. **Explicit feedback** is provided whenever users accept or reject a given agent's suggestion by using the respective buttons ✓ / ✗. **Implicit feedback** is provided automatically by the system when suggestions are rejected because they have been ignored for more than 10 document inserts.

Contexts to Learn From – To maximize the amount of knowledge extracted from a single user interaction, agents consider two distinct points in time: the *trigger point* and the *interaction point* of a given suggestion. The **trigger point** is the point in time at which the guidance agent decided to *trigger* a new suggestion. Figure 6 shows that the model at the trigger point is the common ancestor between the original model and the suggested model. As the context vectors at the trigger point are responsible for an agent making a suggestion, updating them directly influences whether the agent will provide a suggestion in a similar analysis state again. The **interaction point** is the point in time at which the suggestion was accepted (or rejected). While the context vectors at the trigger point were responsible for initially proposing the suggestion, the suggestion was still relevant at the point where the user accepted it. Consequently, suggestions similar to that made at the trigger point could also become relevant in the future, leading us to update the respective context vectors as well.

Weighting of Updates – Depending on the type of *point in time* and the *type of feedback* that was provided, we update the context vectors with different weights. The weights are outlined in the adjacent table and have been determined experimentally, based on experience from previous work [EASS*18]. Weights for explicit feedback are twice as high as weights for implicit feedback. This ensures that agents primarily learn from explicitly provided feedback but do not ignore the associated implicit information. Furthermore, it gives users the possibility to manually overwrite previously learned rules when necessary, e.g., after changing the analysis direction. **Intermediate points** encountered between the trigger and acceptance or rejection are not updated.

	Explicit	Implicit
Trigger Point	6	3
Intermediate	0	0
Interaction Point	2	1

7. Evaluation

Since the evaluation of mixed-initiative systems depends on a variety of interlinked aspects, a systematic assessment of our technique as a whole is challenging. Therefore, we provide a multi-faceted evaluation. We first report the results from a qualitative expert study to showcase the usability of the provided system and its guidance agents. In our quantitative evaluation, we utilize three approaches to assess the effect of our technique. We determine the effects on result quality in a topic ranking task, verify whether the contexts identified by the agents correspond to human-selected optimizations, and confirm that contexts are transferable between similar datasets.

7.1. Qualitative Evaluation: Expert User Study

This evaluation aims to gain insights into how co-adaptive guidance impacts the topic modeling process. We investigate three main questions: (1) What is the domain expert's preconception of guided analysis processes? (2) How does the visual analytics interface support interactive model optimization? (3) What is the impression on the utility of the provided co-adaptive guidance? We have since used this study's feedback to improve our approach. In particular, we optimized the layout by displaying the Sankey next to the icicle, visually simplified the agent cards, and replaced the log-likelihood ratio driven topic descriptor extraction with YAKE [CMP*20].

Methodology – Each session of our pair analytics study [KF14] was a recorded video call with a domain expert and lasted 120-150 minutes. All sessions were split into three phases: introduction and elicitation of expectations (20 minutes), system use with intermittent questionnaires (45-70 minutes), and a semi-structured interview.

Participants – We recruited four participants (2F/2M) from political science (**P1**–**P3**) and computational linguistics (**CL1**). They are Ph.D. students (P1, P3, CL1) and postgraduate researchers (P2) and have previous experience optimizing topic models. While P1 and P3 were interested in getting a broad and detailed overview of the data, respectively, CL1 was interested in which topic was occupied by which speaker in order to analyze framing effects later.

Dataset – All participants refined an IHTM of the second 2012 Obama-Romney debate, as they all had experience with analyzing political debates, but not with this particular debate prior to the study.

Expectations towards Guidance – Participants noted that the expected model results depend on the envisioned downstream task (P1 & P3) and decided to optimize for broader (P1 & P2) or more fine-grained (P3 & CL1) results. CL1, who had previously manually op-

timized topic models, was sure that the results she could achieve using visual analytics *"will surely be better and get closer to what my corpus is actually like*". P3 appreciated guidance but noted the importance of human control in the process. He explained that guided analysis should only make "testable" suggestions – a requirement that our system fulfills with preview and comparison views. P3 further expected that he would get annoyed quickly if the agents did not do what he wanted. In such a case, he would prefer for them to "*just let me do it on my own*".

Participants quickly grasped the agents' adaptive nature, comparing them to their changing feeds on social media platforms. Drawing from this experience, P3 was worried that the system would not "*recommend new options*" and put too much emphasis on the initial interactions without providing controls to override this shortcoming or reverse a previous interaction. He appreciated that our system only uses certain rules above a weight threshold as an approach to preventing this behavior. P3's concern raises the question of balancing *exploration* of the potential model state-space with the *exploitation* of the expressed preferences. Most participants were aware of the risk of creating a "*bubble*" (P3) in which models would also learn the analyst's inherent biases (P1).

Design and Usability – All participants frequently work with larger amounts of text semi-manually. P1 and P2 expressed that they would typically "*spend a lot of time reading documents*" (P1). P2 felt that it was difficult for her to make well-grounded decisions based on the set of keywords and instead fell back to reading the underlying text before making a decision. In general, all participants found the topic labels not always immediately obvious, with CL1 saying "*the keyword extraction is bad*". We had initially aimed to mitigate this issue by providing more details on demand but have since exchanged the keyword extraction algorithm. During the study, visual analytics novice P2 adapted her mental model from being overwhelmed ("*hard to grasp all this*") to perceiving the system as "*more convenient and intuitive*" and having "*the curiosity to dig deeper*".

Guidance with Agents – When an agent re-suggested a rejected optimization, P2 noted that the suggestions would not provide "*enough diversity*". We are now preventing agents from immediately repeating a rejected suggestion. P1 felt "*there are no heuristics*" to distinguish the quality of simultaneous suggestions since he had not noticed the certainty-based opacity changes to the preview icicle. A future version of the system could display which agents align best with user preferences and which ones are more exploratory. P2 proposed introducing agent-specific "*training sessions*" to update the agents in batches rather than training them during the model optimization.

As participants were aware that the agents learned from their interactions, they often traded off between teaching the agents and progressing their refinement. CL1 debated how much time to spend reviewing unexpected suggestions, trading off her immediate analysis goal and the model profiting from her feedback when spending time to review the suggestion carefully. Similarly, P1 kept rejecting unwanted suggestions, saying that providing feedback was "*not helpful for me, but helpful for the model.*" When P3 encountered a suggestion that did not fit his analysis plan, he instead ignored it and let it decay over time to avoid teaching something wrong and later spending much time on re-training.

Guidance Timing and Helpfulness – While using the system, participants were asked to rate the timing and helpfulness of suggestions on a scale from 1 (bad) to 5 (good) three times in fixed intervals. On average, they rated the timing 3.75, 3.75, and 4.25 at the three time points (overall average 3.83, SD 0.72). Helpfulness was rated 3.75, 4, and 3.75 (average 3.92, SD 0.67). While we observe an upwards trend in timing ratings as the analysis progresses, more participants in a more controlled study are needed to confirm the significance of this trend. Both P1 and CL1 stated that they found it difficult to judge the extent to which the system had learned from their interaction. However, the quantitative evaluation suggests that agents, in fact, learn tangible knowledge from interactions. Consequently, further research is needed to investigate how context adaptation of the agents can be translated to more readily noticeable system behavior changes.

7.2. Quantitative Evaluation: Model Quality & Transferability

The expert user study showed the applicability of our approach and found that users liked interacting with their agents to train them. To evaluate the agents' impact on the resulting topic model, we performed three quantitative experiments targeting different aspects.

Datasets – Throughout all evaluations, we use two datasets: the first (DB1) and the second (DB2) presidential debate of Obama and Romney in 2012 [CNNb, CNNa]. We chose the second debate because we had used it in the expert user study and the first debate to verify that the learned contexts are transferable to similar datasets.

7.2.1. Topic Ranking Task

Our previous work on the automatic refinement of topic models has shown that optimizing models towards improved quality metrics actually worsens their *perceived quality* when compared to a baseline [EASD*19]. To verify the validity of the agents' learnings, we repeated our previous perception experiment and let study participants rank optimized and unoptimized model results.

Methodology – We first prepared a *baseline* IHTM result for both the first and the second debate. We then trained the agents through the guided refinement of the second debate (*supervised* result). The agents' suggestions were manually reviewed by a topic modeling expert and accepted or rejected following our previously presented methodology of aiming to match an established, expected topic distribution [EASD*19]. We extracted the agents' learned user preference models and used them to perform *automatic* optimizations on both debates, where we accepted all presented suggestions. Whenever two agents made a suggestion at the same time, we selected the agent with higher confidence and resolved draws using random picks. We use the five obtained modeling results in two independent ranking tasks between the three results for the first debate and the two results from the second debate. Participants were asked to rank the results (given as lists of topic descriptors) on a scale from 1 (best) to 3 (worst) and 1 to 2, respectively, and to justify their decision.

Participants – In addition to three participants from the qualitative evaluation, we recruited five new ones (overall: 5F/3M) with various backgrounds (computational linguistics, political science, computer science) to avoid biased results based on previous involvement.

Results – The results from 40 ranking annotations are summarized in Table 1, and show that, on average, automatic refinement

Corpus	Type	Rank (SD)
Second Debate [CNNb]	baseline	2.0 (.93)
	automatic	1.9 (.64)
	supervised	2.1 (.99)

Corpus	Type	Rank (SD)
First Debate [CNNa]	baseline	1.9 (.35)
	automatic	1.1 (.35)

Table 1: *Average annotator rank for different models from two corpora. Users consistently preferred models automatically optimized through agents over the baseline.*

Type	Corpus	Combine	Split	Small	Outlier	Chains	Worst
Training	DB2	120 (27)	103 (16)	10 (7)	98 (12)	13 (2)	142 (18)
Automation	DB2	139 (19)	100 (25)	13 (1)	120 (35)	17 (2)	154 (38)
Automation	DB1	118 (15)	60 (14)	9 (1)	110 (29)	7 (0)	107 (24)

Table 2: *Counts for contexts seen (applied) are similar during training, automation on the same dataset, and transfer to unseen data.*

was received better than the baseline with a rank of 1.9 and 1.1, respectively. This constitutes an improvement over our previous work where the baseline clearly outperformed automatic refinement [EASD*19] and suggests that the **contexts learned by agents are better suited for suggesting refinements** than simply optimizing towards quality metrics. From the participants' comments, we conclude that they either preferred results with few topics (*supervised* result) or with many topics (*baseline* result) for the second debate. As a result, we observe a high standard deviation in the ranking, and no model gains a clear advantage. For the first debate, the ranked results had similar numbers of topics, and participants focused more on content-features (mentioning keywords in their justifications). As justifications for their preferences, participants particularly pointed out the better separation of topics and their cohesive granularity.

7.2.2. Contextualization of Agents

The ranking task confirms that agents can learn context rules that can be used to refine models automatically. In this experiment, we aim to assess to what extent the contextualizations learned by agents matches user expectations in a given modeling scenario.

Methodology – We sampled nine intermediate modeling states from the *automatic* refinement processes of the first and second debate in which either a *combine*, *split*, or *reinsert worst* optimization was suggested by the respective agents, and one state in which none of the agents suggested an optimization. For each state, participants were provided with a screenshot of the icicle plot (annotated with topic IDs) and a list of topic descriptors and asked to specify which optimization (combine, split, reinsert worst, or no action) they would have performed (on which topics), justify their decision, and give their certainty on a scale from 1 (uncertain) to 3 (certain). We only selected optimizations from the three most active agents to avoid overwhelming users.

Participants – The task was performed by the same eight participants that produced the result ranking annotations.

Results – Participants produced 70 context annotations with an average certainty of 2.62 (SD 0.58). Most frequently, they chose to reinsert topics (23); all other operations were almost equally distributed (15 combine, 16 split, 16 no action). There were minimal differences between the agent's prediction accuracy for the three optimizations. On average, agents successfully predicted the operation that participants wanted to perform in 35% of cases. For the three participants from computer science, this value increased to 56%. Considering that a computer scientist trained the agent rules, those results suggest that different user groups might need substantially different agents. As one of our participants pointed out, a common issue in modeling discourse data is the creation of a so-called *moderation*-topic. She noted that "this topic does not seem to have any specific content, and many of the descriptors are only expressions for discourse transition." In many cases, our participants focused on trying to optimize this topic. This is particularly apparent in participants from computational linguistics, where 47% of responses are targeted at splitting or removing this specific topic. On the other hand, computer scientists consistently ignored the moderation topic and focused on other optimization targets.

The evaluation also shows that all but two participants had very skewed preference distributions for splitting or reinserting topics, primarily picking one or the other. However, users from both groups reference the same topics in their justification as both operations intuitively have a similar effect. Artificially removing the difference between operations would increase the overall accuracy to 52%. We suspect that users from both groups would train similar contexts for the split or combine agent, respectively, and achieve similar final modeling results. These initial findings highlight the importance of personalized and adaptive guidance and suggest that **different user groups have differing preferences**.

7.2.3. Contexts Learned and Rule Transferability

Our last experiments investigate the transferability of the learned user preference models between datasets. Such transferability is the foundation for continuous agent training across users and corpora as it shows that agents can learn generalizable rules rather than overfitting on specific analysis contexts.

Methodology – Through an exploratory analysis, we aim to verify the finding that learned user preference models can be transferred between datasets. Before presenting the results, we define two core terms: A context is *seen* if it is closed and *considered* for classification by the agent at least once. A context is *applied* if it is seen and *has been used* in the classification at least once. Recall that only those contexts with a sufficient weight are used in this decision.

Results – First, we investigated how many contexts were seen and applied during agent training. Table 2 shows that agents saw up to 142 contexts and, on average, applied 24.72% of contexts at least once. Agents with a larger impact on the result (e.g., combine and split) are accepted much more frequently and learn significantly more contexts. The table also shows that comparable amounts of contexts are seen when automating refinement on the same or similar datasets. Here, agents apply 18.66% and 15.99% of seen contexts.

Next, we were interested in verifying that the similar numbers actually manifested in the same contexts being encountered. Table 3 shows the percentage of encountered contexts during automation were also encountered during training. All agents saw at least 48.54% of training contexts (split agent). This number increases with an increasing ratio of seen and applied contexts in training, highlighting that confident rules are more likely to be transferred. As a result, the *remove chains* agent that learned many unconfident contexts was not able to make a suggestion in automation. In contrast, both the

Context	Corpus	Combine	Split	Small	Outlier	Chains	Worst
Seen	DB2	86.67%	48.54%	90.0%	93.88%	53.85%	64.08%
	DB1	98.33%	58.25%	90.0%	112.24%	53.85%	75.35%
Applied	DB2	48.15%	75.0%	14.29%	216.67%	0.0%	116.67%
	DB1	55.56%	87.5%	14.29%	241.67%	0.0%	133.33%
G^2	DB2	30.0%	15.0%	45.0%	10.0%	15.0%	30.0%
	DB1	30.0%	15.0%	30.0%	0.0%	5.0%	15.0%

Table 3: *Ratios of transferable contexts across two target corpora.*

remove worst and *remove outlier* agents learned confident contexts which were not applied during the training session. As a result, they applied up to 2.4 times as many contexts during automation.

Finally, we use the G^2 [Dun93] metric to determine the most characteristic contexts for each agent. Table 2 shows that up to 45% of characteristic contexts learned in training remain characteristic during automation, suggesting that they capture distinctive, generalizable situations. These findings illustrate that we can **obtain refined user preference models by continuously training agents** over long periods of time and across datasets. However, future research and long-term evaluations are needed to confirm this finding.

8. Discussion

We have presented a technique for the adaptation of guidance based on learned user preference profiles, and implemented and evaluated it in the context of topic model refinement.

Incorporating Semantics – Our evaluation of agent contextualization has shown that participants frequently aimed to optimize one specific topic. As context vectors in our implementation only contain information about metrics and modeling events, they are not ideally suited for capturing those particular interaction patterns. As a first step to addressing semantically different expectations, our implementation provides alternative suggestions for several agents. However, future work could investigate the integration of semantic representations into the context vectorization, thus enabling a more personalized contextualization of guidance.

Teaching Agents – Participants in our expert user study showed a noticeably high motivation to teach their domain knowledge to *their* agents and asserted that optimizing such a guidance model, even over the span of several days, is a worthwhile investment. Even though our system represents the agents as simple cards and not, e.g., animated avatars, the concept of training their personal set of agents seemed to increase engagement in our participants. However, we observed that despite explicitly aiming to teach their agents, users became more susceptible to confirmation bias after a short time of using the system: while some were initially skeptical of the agents' quality, they rapidly grew comfortable in accepting guidance without strict quality assessment. This finding from our study reinforces uncertainty communication and the mitigation of confirmation bias as central challenges for guidance in visual analytics.

Generalizability – In contrast to other existing approaches, our technique relies on a vectorization of the analysis context that captures the temporal development of metrics and events rather than similarity to previous interactions. In particular, it is not specific to topic model refinement but could be generalized to other model building tasks in visual analytics that are supported by quality metrics and existing refinement strategies. A considerable challenge to tailor our approach is the identification of a suitable context vectorization. The presented implementation uses a sliding window and computes the difference between subsequent metric values (i.e., window size = 1). A suitable vectorization for other domains should consider the trade-off between context complexity and both the required amount of training data and the perceived agility of agents during interaction.

Classification Algorithm – In our implementation, the context vectorization relies on diverse topic model quality metrics and IHTM-specific events to capture the model development and encodes this information at different levels of granularity in full and partial context vectors. Agents learn at all granularities and filter context vectors by closedness when classifying to avoid duplicated information biasing the result. As a result, a rule-based classifier and a simple approach to relevance feedback proved effective. It is likely that more complex machine learning techniques might enable even better performance, potentially at the cost of intelligibility, cold-start issues, and implementation complexity.

Limitations – The proposed technique currently requires an initialization with (synthetic) pretraining data to be effective. Even with proposed simplifications like metric shapes, the state space spanned by context vectors is large, and training a classifier from scratch is not feasible with the typical number of interactions in a visual analytics session. The technique's design aims to combat this issue by learning (more) generalizable knowledge on partial contexts. Our evaluation suggests that rules are transferable between datasets, with automatic optimization outperforming the baseline. While future research is needed to confirm this finding, transferable rules would enable continuous training of agents, overcoming this limitation.

In this paper, we provide four distinct evaluation approaches to validate our technique from both human-centered and algorithm-centered perspectives. However, the paper only instantiates the technique for the iterative refinement of topic models. Further investigations are necessary to evaluate how the technique performs on different tasks and over longer periods of time. Additionally, the evaluation can only establish single-objective, context-dependent agents as one possible way to provide guidance. In particular, it does not provide a comparative evaluation against other guidance techniques.

9. Conclusion

We have presented a visual analytics technique for co-adaptive guidance through contextualized preference learning and instantiated it in an approach to topic model refinement. The technique is centered around guidance agents that, over time, learn user preferences to provide distinct guidance suggestions in specific contexts. The flexible nature of the context vectorization enables system designers to choose the metrics, events, or semantic representations that are most relevant to their application scenario. Our initial evaluation within a qualitative user study shows that the technique is accepted and valued by domain experts. The quantitative results show that context rules learned by agents can be used in automatic topic model optimization, outperforming previous optimization approaches. We plan to extend our technique on *contextualizing* guidance to enable agents to personalize the *content* of their suggestions in the future. Our system prototype is available at *topic-model-guidance.lingvis.io*.

Acknowledgements – This work has been partially funded by the

DFG within grant numbers 376714276 and 455910360 (SPP-1999). Open access funding enabled and organized by Projekt DEAL. [Correction added on 08 November 2021, after first online publication: Projekt Deal funding statement has been added.]

References

[AAR*09] ANDRIENKO G., ANDRIENKO N., RINZIVILLO S., NANNI M., PEDRESCHI D., GIANNOTTI F.: Interactive visual clustering of large collections of trajectories. In *IEEE Symposium on Visual Analytics Science and Technology* (2009), pp. 3–10. doi:10.1109/VAST.2009.5332584. 2

[AEK00] ANKERST M., ESTER M., KRIEGEL H.-P.: Towards an effective cooperation of the user and the computer for classification. In *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining* (2000), pp. 179–188. doi:10.1145/347090.347124. 2

[AGH99] ALLEN J. F., GUINN C. I., HORVITZ E.: Mixed-initiative interaction. *IEEE Intelligent Systems and their Applications 14*, 5 (1999), 14–23. 1

[BLBS11] BREMM S., LANDESBERGER T. V., BERNARD J., SCHRECK T.: Assisted Descriptor Selection Based on Visual Comparative Data Analysis. *Computer Graphics Forum 30*, 3 (2011), 891–900. doi:10.1111/j.1467-8659.2011.01938.x. 4

[BNJ03] BLEI D., NG A., JORDAN M.: Latent Dirichlet allocation. *Journal of Machine Learning Research 3*, 1 (2003), 993–1022. doi:10.1162/jmlr.2003.3.4-5.993. 3

[CAS*18] COLLINS C., ANDRIENKO N., SCHRECK T., YANG J., CHOO J., ENGELKE U., JENA A., DWYER T.: Guidance in the human–machine analytics process. *Visual Informatics 2*, 3 (2018), 166–180. doi:10.1016/j.visinf.2018.09.003. 2

[CGM*17] CENEDA D., GSCHWANDTNER T., MAY T., MIKSCH S., SCHULZ H.-J., STREIT M., TOMINSKI C.: Characterizing Guidance in Visual Analytics. *IEEE Trans. Visualization and Computer Graphics 23*, 1 (2017), 111–120. doi:10.1109/TVCG.2016.2598468. 2

[CGM19] CENEDA D., GSCHWANDTNER T., MIKSCH S.: A Review of Guidance Approaches in Visual Data Analysis: A Multifocal Perspective. *Computer Graphics Forum 38*, 3 (2019), 861–879. doi:10.1111/cgf.13730. 2

[CLRP13] CHOO J., LEE C., REDDY C. K., PARK H.: UTOPIAN: User-Driven Topic Modeling Based on Interactive Nonnegative Matrix Factorization. *IEEE Trans. Visualization and Computer Graphics*, 12 (2013), 1992–2001. doi:10.1109/TVCG.2013.212. 2

[CMP*20] CAMPOS R., MANGARAVITE V., PASQUALI A., JORGE A., NUNES C., JATOWT A.: YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences 509* (2020), 257–289. doi:10.1016/j.ins.2019.09.013. 4, 7

[CNNa] CNN POLITICAL UNIT: Transcript: First presidential debate. https://edition.cnn.com/2012/10/03/politics/debate-transcript/index.html. 8, 9

[CNNb] CNN POLITICAL UNIT: Transcript: Second presidential debate. http://politicalticker.blogs.cnn.com/2012/10/16/transcript-second-presidential-debate/. 8, 9

[DFB11] DRUCKER S. M., FISHER D., BASU S.: Helping Users Sort Faster with Adaptive Machine Learning Recommendations. In *Human-Computer Interaction – INTERACT* (2011), Campos P., Graham N., Jorge J., Nunes N., Palanque P., Winckler M., (Eds.), pp. 187–203. 2

[Dun93] DUNNING T.: Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics 19*, 1 (1993), 61–74. 10

[DYW*13] DOU W., YU L., WANG X., MA Z., RIBARSKY W.: HierarchicalTopics: visually exploring large text collections using topic hierarchies. *IEEE Trans. Visualization and Computer Graphics 19*, 12 (2013), 2002–2011. doi:10.1109/TVCG.2013.162. 2

[EAKC*20] EL-ASSADY M., KEHLBECK R., COLLINS C., KEIM D. A., DEUSSEN O.: Semantic Concept Spaces : Guided Topic Model Refinement using Word-Embedding Projections. *IEEE Trans. Visualization and Computer Graphics 26*, 1 (2020), 1001–1011. doi:10.1109/TVCG.2019.2934654. 2, 3, 4

[EASD*19] EL-ASSADY M., SPERRLE F., DEUSSEN O., KEIM D., COLLINS C.: Visual Analytics for Topic Model Optimization based on User-Steerable Speculative Execution. *IEEE Trans. Visualization and Computer Graphics 25*, 1 (2019), 374–384. doi:10.1109/TVCG.2018.2864769. 2, 3, 5, 6, 8, 9

[EASS*18] EL-ASSADY M., SEVASTJANOVA R., SPERRLE F., KEIM D., COLLINS C.: Progressive Learning of Topic Modeling Parameters: A Visual Analytics Framework. *IEEE Trans. Visualization and Computer Graphics 24*, 1 (2018), 382–391. doi:10.1109/TVCG.2017.2745080. 2, 7

[EFN12] ENDERT A., FIAUX P., NORTH C.: Semantic Interaction for Visual Text Analytics. In *Proc. Conf. Human Factors in Computing Systems* (2012), pp. 473–482. doi:10.1145/2207676.2207741. 2

[HBGSS14] HU Y., BOYD-GRABER J., SATINOFF B., SMITH A.: Interactive Topic Modeling. *Machine Learning 95*, 3 (2014), 423–469. doi:10.1007/s10994-013-5413-0. 2

[HC15] HOQUE E., CARENINI G.: ConVisIT: Interactive Topic Modeling for Exploring Asynchronous Online Conversations. In *Proc. Int. Conf. Intelligent User Interfaces* (2015), pp. 169–180. doi:10.1145/2678025.2701370. 2

[HSD19] HOHMAN F., SRINIVASAN A., DRUCKER S. M.: TeleGam: Combining Visualization and Verbalization for Interpretable Machine Learning. In *IEEE Visualization Conference (VIS Short)* (2019), pp. 151–155. doi:10.1109/VISUAL.2019.8933695. 5

[KDEP20] KIM H., DRAKE B., ENDERT A., PARK H.: ArchiText: Interactive Hierarchical Topic Modeling. *IEEE Trans. Visualization and Computer Graphics* (2020), 1–12. doi:10.1109/TVCG.2020.2981456. 2

[KF14] KAASTRA L. T., FISHER B.: Field Experiment Methodology for Pair Analytics. In *Proc. Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization at VIS* (2014), pp. 152–159. doi:10.1145/2669557.2669572. 7

[KLHO19] KOCH J., LUCERO A., HEGEMANN L., OULASVIRTA A.: May AI?: Design Ideation with Cooperative Contextual Bandits. In *Proc. CHI Conf. Human Factors in Computing Systems* (2019), pp. 1–12. doi:10.1145/3290605.3300863. 2

[Lan] LANG K.: Home Page for 20 Newsgroups Data Set. http://qwone.com/jason/20Newsgroups/. 6

[MGH13] MCDAID A. F., GREENE D., HURLEY N.: Normalized Mutual Information to evaluate overlapping community finding algorithms. *arXiv e-prints* (2013), 1–3. arXiv:1110.2515. 6

[MHM18] MCINNES L., HEALY J., MELVILLE J.: UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv e-prints* (2018), 1–63. arXiv:1802.03426. 4

[PKL*18] PARK D., KIM S., LEE J., CHOO J., DIAKOPOULOS N., ELMQVIST N.: ConceptVector: Text Visual Analytics via Interactive Lexicon Building Using Word Embedding. *IEEE Trans. Visualization and Computer Graphics 24*, 1 (2018), 361–370. doi:10.1109/TVCG.2017.2744478. 2

[SBE*18] SEVASTJANOVA R., BECK F., ELL B., TURKAY C., HENKIN R., BUTT M., KEIM D. A., EL-ASSADY M.: Going beyond Visualization : Verbalization as Complementary Medium to Explain Machine Learning Models. In *Workshop on Visualization for AI Explainability at IEEE VIS* (2018), pp. 1–6. 5

[SDCW20] SANH V., DEBUT L., CHAUMOND J., WOLF T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv e-prints* (2020), 1–5. arXiv:1910.01108. 4

[SJB*20] SPERRLE F., JEITLER A., BERNARD J., KEIM D., EL-ASSADY M.: Learning and Teaching in Co-Adaptive Guidancefor Mixed-Initiative

Visual Analytics. *EuroVis Workshop on Visual Analytics* (2020), 1–5. doi:10.2312/eurova.20201088. 1, 2, 3

[SSKEA19] SPERRLE F., SEVASTJANOVA R., KEHLBECK R., EL-ASSADY M.: VIANA: Visual Interactive Annotation of Argumentation. In *Proc. IEEE Conf. Visual Analytics Science and Technology (VAST)* (2019), pp. 1–12. doi:10.1109/VAST47406.2019.8986917. 5

[TvdS15] TEKIN C., VAN DER SCHAAR M.: Distributed Online Learning via Cooperative Contextual Bandits. *IEEE Trans. on Signal Processing 63*, 14 (2015), 3700–3714. doi:10.1109/TSP.2015.2430837. 2

[ZH02] ZAKI M. J., HSIAO C.-J.: CHARM: An Efficient Algorithm for Closed Itemset Mining. In *Proc. SIAM Int. Conf. on Data Mining* (2002), pp. 457–473. doi:10.1137/1.9781611972726.27. 6