The Exploratory Labeling Assistant: Mixed-Initiative Label Curation with Large Document Collections

Cristian Felix
New York University
New York, USA
cristian.felix@nyu.edu

Aritra Dasgupta
Pacific Northwest National Laboratory
Richland, USA
aritra.dasgupta@pnnl.gov

Enrico Bertini
New York University
New York, USA
enrico.beritni@nyu.edu

ABSTRACT

In this paper, we define the concept of exploratory labeling: the use of computational and interactive methods to help analysts categorize groups of documents into a set of unknown and evolving labels. While many computational methods exist to analyze data and build models once the data is organized around a set of predefined categories or labels, few methods address the problem of reliably discovering and curating such labels in the first place. In order to move first steps towards bridging this gap, we propose an interactive visual data analysis method that integrates human-driven label ideation, specification and refinement with machine-driven recommendations. The proposed method enables the user to progressively discover and ideate labels in an exploratory fashion and specify rules that can be used to automatically match sets of documents to labels. To support this process of ideation, specification, as well as evaluation of the labels, we use unsupervised machine learning methods that provide suggestions and data summaries. We evaluate our method by applying it to a real-world labeling problem as well as through controlled user studies to identify and reflect on patterns of interaction emerging from exploratory labeling activities.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

Author Keywords

Exploratory Labeling; Text Analysis; Visualization; Document Labeling.

INTRODUCTION

In many real-world situations analysts are faced with the problem of organizing a document collection around a not yet fully defined set of categories. For example, law enforcement investigators who need to categorize digital communications among criminals and victims according to type of crime or transaction; marketing teams who need to categorize product reviews according to what customers reveals in their comments; or groups of social scientists who need to categorize a large collection of messages gathered from social media according to some particular type of behavior.

In these situations, analysts typically come to this task with a loosely defined idea of what categories to use and, as they analyze more of the data, develop over time a better understanding of what information is contained in the collection and how documents can and should be categorized (a problem similar to what Kuletza et al. call “concept evolution" [25]).

While with a small number of documents it is always possible to perform such type of analysis manually, e.g., by using methods such as those proposed by *grounded theory* [8], when dealing with large document collections (with tens of thousands of documents) some degree of automation and cooperation between the human analyst and the machine is desirable.

One option in this case is to analyze and label only a subset of documents and then use the manually annotated data as a way to “bootstrap" a classifier. This solution however leaves the analyst with a high degree of uncertainty and is prone to inaccuracies due to sampling artifacts. For example, instances of rare, still unknown and yer relevant categories may be lost in the process. An alternative strategy is to use unsupervised learning algorithms such as *topic modeling* or *clustering*. These methods, however, present a few relevant drawbacks. First, the output of these methods is often out of sync with the mental model of the analyst: labels that are relevant to an analyst might not be captured by an algorithm if they are not frequent or discriminative enough for a given collection. Second, the output may be very noisy and hard to interpret, especially with topic modeling [34].

Finally, another option is to use methods inspired by *active learning* [36] where a classifier is developed over time by repeatedly asking a “human oracle" to label individual documents proposed by the model. Such methods however do not work for our case as they presume the existence of a predefined set of labels. One possible solution is to develop variations of active learning such that labels can be defined and refined over time. One such method is *structured labeling* [25], a method that permits users to refine labels over time. The main downside however is its limited transparency: since the user can only provide information by labeling individual or group of instances, it is not always clear what the model has learned.

UIST '18, October 14–17, 2018, Berlin, Germany

ISBN 978-1-4503-5948-1/18/10...$15.00

DOI: https://doi.org/10.1145/3242587.3242596

As a solution to these problems we propose *exploratory labeling*: a mixed-initiative approach to assist human analysts in creating label specifications for large document collections. The main idea behind exploratory labeling is to bypass the problem of labeling individual documents by supporting the user in developing transparent (yet machine readable) specifications that link documents to labels. Rather than asking the user to link documents and labels directly, we assist the user in developing sets of terms that describe the label and then use such descriptions to match documents to labels.

The advantage of such solution is twofold. First, by its very nature it creates transparent specifications for the labels. Second, it lends itself to their progressive and iterative development. Adding or removing terms to labels is in fact easier than shifting whole sets of documents from one label to another as more refined label definitions are developed over time.

More specifically, we characterize exploratory labeling as a three-step process: label *ideation*, label *specification* and label *evaluation*. In label ideation the goal is to generate the labels. In label specification, the goal is to develop a transparent specification the system can use to match documents with labels and human can easily interpret. In label evaluation, the goal is to verify such matching, spot potential issues, and derive insights for label refinement.

Given the complexity of these tasks and the high number of documents involved in the analysis, it is not reasonable to expect users to perform these activities without computational support. For this reason, we also propose mixed-initiative methods to support the user with each of these phases we described. More specifically, we provide methods to suggest new labels to add to the existing pool of labels; to suggest terms to add or remove from a label specification; and to verify the connection between labels and documents.

In order to evaluate our method, we provide a set of studies. We describe a case study about how we used our system to help a group of investigators categorize a large set of emails coming from scamming activities. We describe a user study aimed at evaluating the usability of the system; the level of topic coverage one can achieve with exploratory labeling; and interpretability of label specifications. Finally, we describe our analysis of how the output of exploratory labeling can be used to create training data for classification tasks.

RELATED WORK

The work most relevant to the tasks of exploratory labeling falls under three broad areas: label ideation, human-centered document labeling, and grouping of documents based on themes or topics. In the following, we discuss our contributions in the context of these three threads.

Label Ideation

The process of investigating a given data set to extract a meaningful set of categories out of it has been described in qualitative data analysis methods, and more specifically in *grounded theory* [8]. In grounded theory the analyst performs a close reading of the documents and performs three main type of analysis: *open-coding* to extract concepts and relationships, *axial-coding* to related these concepts together, and *selective coding* to organize the codes into a structure that forms the theory.

The work that we propose here bears some resemblance to this process. In our case, we also want to extract concepts (labels) from documents but our goals and scope are different. The goal of our work is to help people organize the document collection around a number of user-defined categories and possibly use these categories as labels to train classification models with these data, not to build a complex theory or taxonomy. Furthermore, our work focuses also on the need to define a computable and complete mapping between the defined labels and the documents; a goal that is much less prominent in qualitative coding. Finally, we also aim at defining labels from a very large collections, an operation that is normally not possible to perform with the standards of qualitative coding.

It is important, however, to mention that some systems and methods have been developed recently to support the work of analysts who want to perform open coding with the support of computational methods. Drucker et al. [17] developed a system that facilitates documents grouping through group-document suggestions. Chen et al. [9] describe the challenges of using machine learning for qualitative coding and built a tool to support this task. Marathe and Toyama [30] proposed a methods to match documents to codes automatically based on a previously developed codebooks. The method, however, does not provide support for generating the codes. Chandrasegaran et al. [6] presented a system that uses linguistic features and interactive visualization to assist analysts in performing open coding. However, they also rely on users assigning labels to individual pieces of the text and have no recommendation mechanisms to aid the user in performing this task. Finally, *Aeonim* [16] is an interface that allows collaborative coding with the goal of ensuring agreement.

Document Labeling

Labeling documents is a fundamental task in unstructured text data analysis. For this reason many methods provide support by facilitating the direct labeling of documents and suggesting documents to label. Active learning [31, 35] is a methods that keeps the user in the loop mostly in the *label specification* step. Interactive approaches leveraging active learning have been proposed where the user is more in control over the *label ideation* processes [21, 25] and human feedback is incorporated within a trained model. Recently, a framework for visual interactive labeling has been proposed by researchers focusing on three components: labeled data, trained model, and the knowledge gained by an expert through the labeling process [3]. Empirical studies have compared the performance of active learning as opposed to visual interactive learning [2] demonstrating the potential benefits of analytical guidance in the labeling process. Our exploratory labeling concept is inspired by these findings, yet, it is different in the way the user can directly specify their mental model about the collection of documents in the form of terms and simple rules. More specifically, rather than labeling documents directly, users create labels with a machine-readable specification, which in turn can be used to match labels to documents.

Human Intervention in Topic Modeling

Unsupervised machine learning algorithms [21] are generally used to categorize sets of documents under thematically similar groups or *topics*. Traditional metrics are unable to capture the coherence of automatically generated topics [7] thereby signifying the need to incorporate human judgment in topic discovery specification, and validation stages [28]. Two main approaches have been developed to human intervention in topic modeling, in the first approach the user receives an intermediate result, and provide some feedback to the system. Systems like Utopian [11] and TopicLens [24] allow users to manipulate topic modeling generated through *non-negative matrix factorization* by changing keyword weights, adding, removing or splitting topics, or interacting with 2D word embeddings. El-Assady et al. [18] proposed a system that allows users to compare different topic models and provide feedback by selecting which topic better match a document. Researchers have also proposed interactive methods that work in conjunction with latent Dirichlet allocation [14, 15] or clustering algorithms [27, 37] for users to incorporate their feedback in the modeling process.

The second approach is to use algorithms to suggest information rather than modify the data directly. *ConceptVector* [32] provides suggestions based on word embedding models to support the creation of a list of keywords related to a theme called concepts. *VisRR* [10] is another system that provides suggestions, in this case, the system suggests documents based on a current filter. The human intervention that happens in the data preparation phase has also been considered: for example, labeling documents, such that the data can later be used for training classifiers.

One important factor ignored by most of these systems is that often the analyst is not sure of what is in the collection, so even just manually labeling requires the user to first perform an exploratory analysis in order to come up with a list of labels. Our approach of exploratory labeling addresses the shortcomings through a mixed-initiative visual analytic method where we aim to strike a balance between automation and human intervention by using a guidance based approach [5] in all stages of the labeling process.

EXPLORATORY LABELING PROCESS

We define *exploratory labeling* as the process of ideating a set of labels L from a document collection D in a way that such labels capture intelligible concepts extracted from the document collection. For each label L_i we also define the set of terms T that captures the meaning of the label and at the same time works as a machine-readable description.

Exploratory labeling is needed in all those situations where an expert needs to organize a document collection according to a series of categories that are not known a priori. For example, an analyst who wants to understand what customers think about a given product; a social scientist who wants to study human subjects from responses they gave in a survey; or a criminal investigator who needs to organize a series of documents according to the type of evidence they contain.

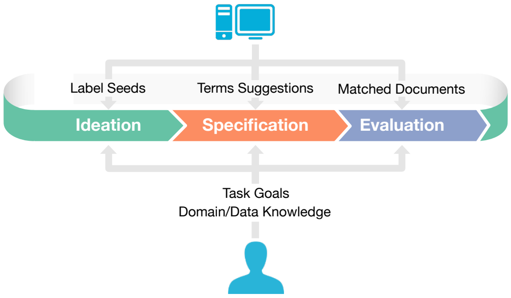


Figure 1. Exploratory labeling: the user is in the center of the label curation process, while the machine provides support in each step for ideation, specification, and evaluation of labels.

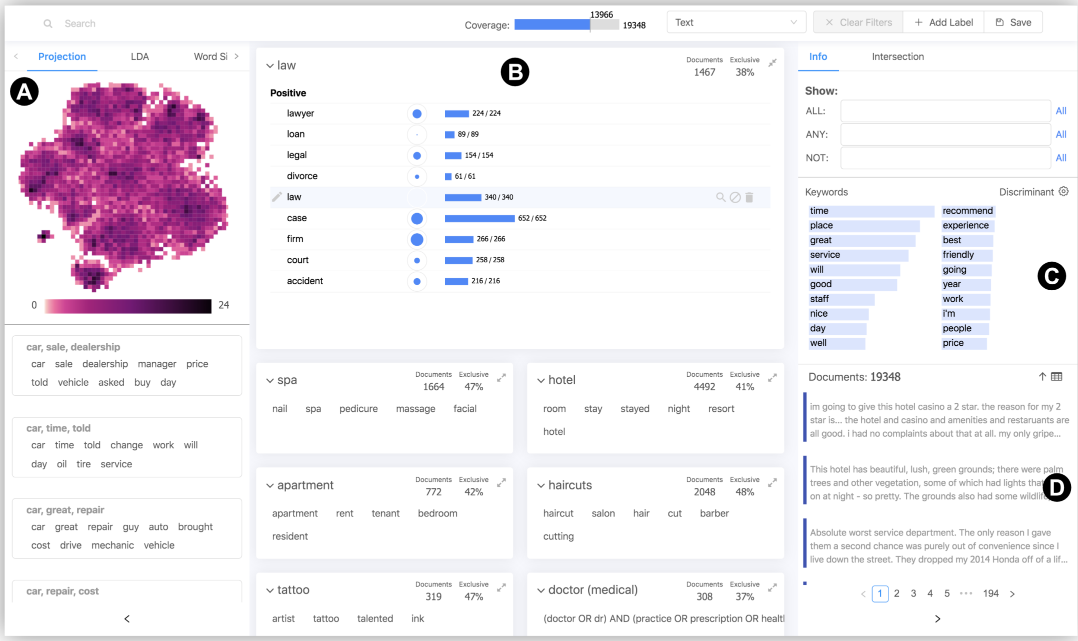
A recurring theme in all of these situations is that the categories and labels the analyst wants to create are not necessarily known a priori, implying that they need to be discovered and refined as the analysis unfolds. In this paper, we tackle this problem by proposing an exploratory process (with an interactive data analytic tool built upon it) that helps human analysts curate labels from document collections.

The process is characterized by three main steps (Figure 1): *label ideation, label specification, and label evaluation*. In **label ideation**, the users' main goal is to come up with ideas about what label they may want to define and may exist in the collection. In **label specification**, the main goal is to build a description of each label using a set of terms found in the collection and potential rules that tie these terms together. In **label evaluation**, the main goal is to verify that a label's specification is able to capture the concept correctly.

These three steps are typically repeated in an iterative fashion by the users to progressively build a set of labels and specifications that satisfy their needs. Common goals in the exploratory labeling process include the need to: uncover as many concepts as possible; build a set of labels that are as distinct as possible, that is, the concepts captured by the labels are well separated; capture concepts that do exist in the document collection (that is, they have statistical support); and create label specifications that are intelligible and semantically coherent.

To guide the user in the achievement of these goals, we propose a mixed-initiative methodology [23] in which the user is the main driver behind the exploratory labeling process. Contrary to most existing paradigms in which labels are first created automatically or semi-automatically by some algorithm (e.g., through topic modeling or clustering) and then are modified by the end-user, we propose a solution in which labels can only be created by the explicit and direct intervention of the user and machine-driven recommendations are used as a supporting tool [13] for the ideation, specification, and evaluation steps.

More precisely, we expect computational methods to support these steps as follows: i) in ideation, by proposing "seed terms" to use to take inspiration for label creation; ii) in specification, by providing suggestions on possibly relevant terms to add to the specification; and iii) in evaluation, by showing documents that match and do not match the current specification.


Search
Coverage: 13966 19348
Text
Clear Filters
Add Label
Save
Projection
LDA
Word Si
A
0
24
car, sale, dealership
car sale dealership manager price told vehicle asked buy day
car, time, told
car time told change work will day oil tire service
car, great, repair
car great repair guy auto brought cost drive mechanic vehicle
car, repair, cost
law
B
Documents 1467 Exclusive 38%
Positive
lawyer 224 / 224
loan 89 / 89
legal 154 / 154
divorce 61 / 61
law 340 / 340
case 652 / 652
firm 266 / 266
court 258 / 258
accident 216 / 216
spa Documents 1664 Exclusive 47%
nail spa pedicure massage facial
hotel Documents 4492 Exclusive 41%
room stay stayed night resort hotel
apartment Documents 772 Exclusive 42%
apartment rent tenant bedroom resident
haircuts Documents 2048 Exclusive 48%
haircut salon hair cut barber cutting
tattoo Documents 319 Exclusive 47%
artist tattoo talented ink
doctor (medical) Documents 308 Exclusive 37%
(doctor OR dr) AND (practice OR prescription OR healtl
Info
Intersection
Show:
ALL: All
ANY: All
NOT: All
Keywords
Discriminant
time place great service will good staff nice day well
recommend experience best friendly going year work i'm people price
C
Documents: 19348
D
1 2 3 4 5 ••• 194

im going to give this hotel casino a 2 star. the reason for my 2 star is... the hotel and casino and amenities and restaurants are all good. i had no complaints about that at all. my only gripe...

This hotel has beautiful, lush, green grounds; there were palm trees and other vegetation, some of which had lights that on at night - so pretty. The grounds also had some wildlif

Absolute worst service department. The only reason I gave them a second chance was purely out of convenience since I live down the street. They dropped my 2014 Honda off of a lif...

Figure 2. ELA: Exploratory Labeling Assistant. Ⓐ Label discovery panel displaying the projection and a list of recommendations. Ⓑ Label specification panel, displaying a set of labels created by the user as cards. ⒸTerms recommendations, providing suggestions of terms that may be relevant to the user. ⒹDocument matching panel, showing the list of documents.

The main reason behind our proposed paradigm is that we believe it is important to develop transparent methods in which end-users are the main driver of the label exploration process. More specifically, we find that existing paradigms, in which algorithms create models first and then humans are expected to interpret and modify them in order to obtain the results they need, do not sufficiently leverage human knowledge and reasoning skills and ultimately may also hinder rather than enhance human expression.

While it may rightly be argued that an excessive manual involvement may make the process too laborious and even prone to biases, we believe it is important to study such type of solutions and verify the extent to which these worries are warranted. In the following section, we first describe the interactive tool we developed to realize the ideas expressed above. Then we devote the rest of the paper for analyzing how well the paradigm we propose works in practice and offer a few metrics to reflect on how effective the whole process is.

ELA: EXPLORATORY LABELING ASSISTANT

In this section, we introduce ELA[1] (Figure 2), the Exploratory Labeling Assistant, an interactive system we implemented to experiment with and validate mixed-initiative label curation process. The user interface (Figure 2) is organized around four main linked views: Ⓐ the *label discovery panel* on the left, which contains tools to support label discovery and ideation (more details below); Ⓑ the *label specification panel* in the center, which hosts visual "cards" representing the labels generated by the user and containing the terms used for their specification; Ⓒ the *terms recommendation panel*, which displays, for a selected label, recommended terms users may want to add to the specification; Ⓓ the *document matching panel*, which contains information about documents that match the specification of one or more selected labels.

These main elements of the user interface support the following workflow. Users generate new labels either by creating them manually (clicking on the "Add Label" button) or by using any tools found in the label recommendation and discovery panel. When a new label is created, it is visualized as a new card in the label specification panel, which contains all the cards/labels created by the user so far. Each card contains an editable title on the top part and an editable list of terms, which represents the specification of the label. At any given time, the user works mainly on two main tasks: creating or deleting labels and adding or removing terms that form the specification of existing labels. To support iterative refinement of label specifications, the term recommendation panel provides recommendations for terms the user may want to add to a selected label. Finally, the label evaluation panel can be used to check which documents match the specification of a selected label.

Label Discovery

The label discovery view contains interactive functionality to help the user ideate new labels. In its current implementation, the view allows the user to use three main methods, *projection*, *topic modeling* and *semantic modeling* for discovery, which are explained in detail below. Each method is shown in a different tab panel allowing the user to chose which method is more suitable for the task in hand. We provide multiple methods because each method emphasizes different aspects of a document collection and their combination can help users discover a higher number of concepts and labels. In particu-

[1] http://exploratorylabeling.com

lar, projections help segment the collection into disjoint sets of topics; topic modeling helps find multiple topics within and among all documents; and semantic modeling helps find clusters of terms that are semantically related. Despite using different algorithms, the methods are designed in a way to produce the same output format, which consists of a list of keyword sets, as shown in Figure 3 Ⓐ. Each method shows a different list to the user. Starting from this list the user can create a label by dragging in the specification panel the whole keyword set or select only a subset of the words and create a new label only with them.

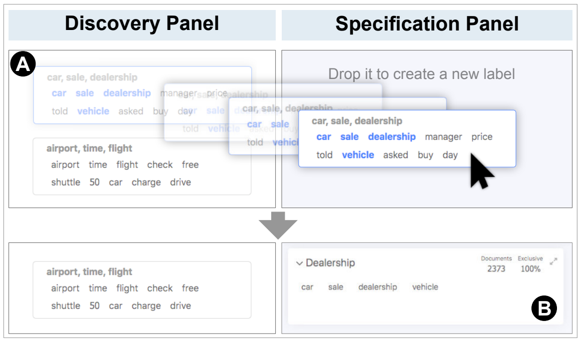

Discovery Panel
Specification Panel
A
car, sale, dealership
car sale dealership manager price
told vehicle asked buy day
airport, time, flight
airport time flight check free
shuttle 50 car charge drive
Drop it to create a new label
car, sale, dealership
car sale dealership manager price
told vehicle asked buy day
airport, time, flight
airport time flight check free
shuttle 50 car charge drive
Dealership
Documents 2373
Exclusive 100%
car sale dealership vehicle
B

Figure 3. Drag-and-drop operation as a way to create a label from a recommendation. Ⓐ Two recommendations provided by the system, Ⓑ the card representing the newly created label using only the selected terms (blue).

Projection

The projection method supports label ideation by creating a multidimensional projection view of the document collection. For ELA we first transform each document into a vector representation using a doc2vec model [26] and then we use the learned representation to project the documents using t-SNE [29], a popular projection algorithm.

In order to make the projection more scalable and readable, we use a density visualization method based on a binning strategy, which enables us to visualize density rather than every single document in the collection (Figure 2 Ⓐ). The rectangular area is divided into small bins which we color with a color intensity scale to represent the number of documents falling into each bin.

In order to generate the keyword sets from the projections, we follow a multi-step process. First, we detect high-density areas to focus on prominent clusters of documents and then we use the documents found in these high-density areas to extract relevant keywords that describe their content.

To detect high-density areas we use a simple procedure: (1) we start from the highest density bin and select all the neighborhood bins at distance d; (2) we use the selected bins to create a document cluster; (3) we remove the bins used in the previous steps, look for the new highest density bin and repeat the same process until the whole set of bins has been processed.

Once the document clusters have been defined, we derive the keyword sets by selecting discriminant terms for each of them. For this purpose, we use a variant of the common term frequency over document frequency measure (TFIDF), proposed by Cilibrasi and Vitanyi [12]. With such measure, we extract the top-n most relevant terms and create the keyword sets.

Topic Modeling

Topic modeling methods aim at automatically discovering from a document collection a number of user-defined topics, where each topic is described by a selected set of terms. In our system, we use the classic Latent Dirichlet Allocation algorithm, also known as LDA [22], to create the keyword sets, where each set just corresponds to each topic extracted by LDA. As described above, the keyword sets (which correspond to the topics in this case), can be used to create new labels in the specification panel.

Semantic Modeling

The semantic modeling algorithm aims at producing keyword sets where the keywords have a close semantic relationship, providing an alternative view of the collection based on word semantic rather than how words are distributed in the document collection.

To produce the keyword sets, we use a word embedding model that learns semantic similarity between words by looking at their context [33], i.e., words that are surrounded by a similar set of words are closer together than those which have more dissimilar contexts. With such embedding, it is possible to calculate the “semantic distance” from any pair of terms contained in the collection. In ELA, we use a pre-trained embedding (generated from the Wikipedia and Gigaword data sets [33]) and generate the keyword sets as follows: (1) we extract the top 1000 most frequent keywords in the dataset; (2) starting from the most frequent one we select all the terms that are at a distance less than a predefined parameter d_s (which can be tuned by the user) and create a keyword set with it; (3) we remove all the terms used in the previous steps from the pool of possible candidates and repeat the previous steps for the next item in the frequent keywords list.

Label Specification and Terms Recommendation

The *label specification panel* shown in Figure 2 Ⓒ is the area where the user creates labels and their specifications. Each label is represented by a card which, in its default state, displays an editable title and the terms included in its specification.

Each label produces a set of connected documents through a matching and ranking function specifying which ones among the existing documents, belong to its set. The sets are not mutually exclusive. Therefore different labels can, in principle, share some of the documents. For this reason, each card also contains information about two metrics: the number of documents matched by the label and the percentage of documents, among those that match, that are exclusive for the label.

The matching mechanism is driven by a function that scores the documents according to how many terms are included in the document. More precisely, for a given label L with terms $T = \{t_1 \dots t_k\}$ included in its specification and a document d, the document score is calculated according to the following function: $match(T,d) \times \sum_{i=1}^{k} tf(t_i,d) \times idf(t_i)^2$, where $match(T,d)$ is the number of terms in the specification that are also present in the document, $tf(t_i,d)$ is the frequency of term t_i in document d, and $idf(t_i)^2$ is the squared frequency of term t_i in the entire collection.

Intuitively, the scoring function scores the documents according to how many terms in the specification are contained in the document (*match* function), weighted according to how specific and frequent these terms are for the document (the rest of the equation). In its default state, a document matches a specification if its score is bigger than zero. More stringent parameters can be used by defining a threshold higher than zero for inclusion.

The user can obtain more details about the label by expanding the card. The expanded version of the card shows additional information about how documents distribute across the terms. By hovering a term, the system displays a bubble next to the other terms to depict information about how many documents they share. This information enables the user to assess the level of redundancy by identifying terms that are highly correlated. The user can also add terms manually using a menu accessible from the top-left corner of the card.

While so far we described the structure of a specification as containing exclusively a simple list of terms, ELA allows the user to define more complex rules in place of a single term. A term can be substituted by a more complex rule tying together multiple terms with a combination of *and*, *or* and *not* logic statements. This is useful in situations when the user needs to define more precisely how to filter the document collection with a given label.

As an additional form of support ELA also provides a *terms recommendation* function which suggests potentially relevant terms to add to an existing label. When a label is selected, the recommendations are displayed on the right-hand side of the user interface, above the document list (Figure 2 X). The suggested terms are displayed in a column layout with bars depicting the frequency of each term following guidelines of Felix et al. [19]. Users can switch between two types of keyword sets, the *most frequent* or the *most discriminant* (according to a relevance score [12]) and they can select terms from this set and add them to the selected label.

Document Matching

The *document matching panel* (Figure 2 Ⓓ) displays for a selected label the list of documents that are retrieved through the scoring function we presented above. The list can be sorted in ascending or descending order according to the score value. Both sorting methods are useful since they enable the user to quickly jump to documents that are scored very high or very low; which in turn permits to reason on how to improve the label specification (by removing or adding terms or entire rules if needed).

Each document in the list is represented by a small snippet, which can be expanded on demand. The snippet is extracted from the document in a way that at least one of the keywords used in the currently selected label will be present and highlighted. When the user clicks on the snippet, a new pop-up window is opened to show the content of the document in full details (including its metadata if available).

In order to verify how many documents the current set of labels share, and thus assess their uniqueness, the user can click on the "intersection" tab and display a correlation matrix. The matrix shows through color intensity how many documents are shared between every possible pair of labels.

CASE STUDY

To showcase how our method can help analysts in a real-world scenario, we describe how ELA has been used in a data analysis project we developed in collaboration with, Agari, a cybersecurity company. Agari aims at protecting their clients from scamming activities, and as part of their endeavor, they obtained a collection of over $60,000$ emails from the email boxes of a network of scammers. The collection contains communications between scammers and victims as well as emails exchanged among scammers on the same network. The goal of the analysis was to understand what kind of strategies scammers use to perpetrate their crimes and, through this activity, categorize emails in categories of scam types.

In this projects we faced three relevant challenges: the analysts had only partial knowledge of what type of labels (scam type) may exist; the data was confidential, which implied the company could not rely on crowdsourcing to help with the labeling; and expert knowledge was required to understand scam types and generate meaningful labels.

Prior to our collaboration, the company had already started a process of identifying types of scams as well as keywords related to the scams types. The company provided us with 3 types of scams they had identified already and between 6 to 12 keywords they found related to each type. Our goal was to use this input to verify the accuracy of the scam types already identified and uncover new categories of scams.

Performing Exploratory Labeling Tasks

As a pre-processing step, we first grouped emails between contacts into one document we called *conversation*. Each conversation contained a concatenation of all communications between two contacts; a strategy suggested by the Agari team and that worked well in their past analyses.

Next, we loaded the dataset into ELA and in partnership with investigators from the company, we conducted an exploratory labeling session with the goal of identifying new types of scams and creating specifications for each identified label. We started by manually adding the labels provided by the company as well as the keywords they had already associated the label with. Then, we looked at the documents matched by these labels, and through this exploration, we realized that some of the keywords were too generic. For example, in a type of scam called "romance scam", the keyword "wonderful" retrieved too many cases in common with "rent scam", another common type of scam (e.g., cases where people included sentences like "the house was wonderful"). Using this method, we identified all the keywords that generated this kind of problem and removed them from the specifications.

In addition to removing terms that were too generic, in some cases, we also had to include more complex rules to disambiguate between the two cases. For example, "romance scams" and "rent scams" had an overlap because both types of scams often mention rooms or other house-related terms in their messages. To handle this problem, we often added negation rules that contained specific bi-grams such as "rental application" in the specification of the romance label.

Once the labels provided by the company were matching the correct set of documents and had little overlap, we aimed at discovering new types of scams. We first configured the system to show only documents not covered by the existing labels and used the projection (Figure 2 Ⓐ) and terms recommendation (Figure 2 Ⓑ) functions to identify new labels. Each new label, once identified, was added to the specification panel and validated using the document matching view. Once we stopped finding relevant information in the recommendations, we started exploring the term recommendations for each label and matching documents to see if we could find other potentially relevant labels the system was not able to recommend.

After producing a satisfactory set of labels and their specifications, we started working on reducing the amount of overlap between the labels. To this purpose, we used the intersection matrix to focus on labels with a high amount of overlap, and then we used the keyword summaries and matching documents to develop better specifications. Once keywords causing excessive overlap were identified we either removed them or added more complex rules able to disambiguate between the cases. In this phase, we frequently used the "exclusive metric" displayed in the cards as a way to measure the exclusivity of the labels. Once this metric for each label was larger than 95%, and we could not identify more than a handful of miss-classified documents we decide to stop.

At the end of this process, we were capable of identifying a total of 12 different types of scams, 9 more than what the company provided initially to us. The number of conversations for each label had a large variance, with one label matching only 19 conversations and others matching up to 626 conversations.

To evaluate the results, we selected a sample of 100 documents, for each label that matched more than a hundred documents, and the whole set of documents for labels with less than a hundred documents. We then manually processed all the sampled documents to count the number of misclassified documents. At the end of this process, we obtained an average precision of 0.94, with 10 out of 12 labels having a precision above 0.95. With two of the remaining labels, we obtained poorer results, respectively 0.77 and 0.79 accuracy values. These were the "rent" and "romance" scams, which, as we have described above tend to have many terms in common.

Lessons Learned

A useful outcome of the case study we described above is a number of lessons we have learned during the development process. We report these lessons here because they reveal useful insights about advantages and challenges of the exploratory labeling process.

Not all documents are relevant. It is important to keep in mind that in many collections not all documents are relevant for the task. In the Agari's case study, for example, our method matched a total of about 68% of the documents contained in the data set. By investigating the other 32%, we found that they were completely irrelevant and not related to any type of scam activity.

Low-frequency labels may still be relevant. In our analysis, we identified a label that matched only 0.32% of the valid documents. But this label was deemed very relevant by the investigators because it was associated with a new and well-defined type of scam. Furthermore, since Agari will keep analyzing data collected in the future, these small frequency label may become more relevant as scamming activities evolve.

Different labels pose different challenges. In this dataset, there was a large variance in the degree of complexity of the specifications required to match the relevant documents in different labels. Some labels needed very simple specifications to match documents correctly, while others needed more complex rules. One example was the scam type "mystery shopper". This scam type would always mention a few variations of the terms "mystery", "secret", and "shopper". Therefore with only a few rules, we were able to match this document with high precision. Other labels like "romance" scams posed a bigger challenge because the conversations were long and the common word much less discriminative.

Other text fields are important. When creating specifications for the labels, we realized that in some cases, the subject line of the emails contained more discriminative information than the body. We designed ELA in a way to allow specifications to use multiple fields at the same time. In cases where we identified more discriminative terms in the subject lines, we created rules that would match only the subject line rather than the body. This capability made some of the specifications much more effective than if we had to rely exclusively on one single source at a time.

VALIDATION

In this section, we present results of the experiments we performed as a way to better understand the advantages and limitations of the exploratory labeling task and the user interface used for the task. More specifically, our goal was to understand: (1) the relationship between the complexity of the task and human performance; (2) the quality of the task outcome in terms of coverage, i.e., the amount of correct labels users are able to find when ground truth is available; and (3) the degree of interpretability of the generated specifications. We also investigate the usefulness of the outcome of the labeling process and how one can leverage it to automatically label new documents.

Methodology

In this experiment, we aimed to reproduce a real-world task in a controlled environment. We created a scenario where participants would pretend to be developers for a website with the primary task of identifying and specifying business types based on a collection of reviews obtained from its users. That is, their goal was to produce labels, where each label would represent a business type, knowing that these labels would be used to create a menu in the fictitious website. The rules created on the specification of each label would be used to

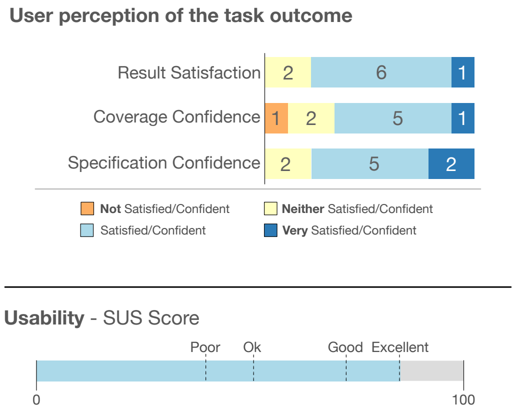


Figure 4. User perception of the task outcome and SUS score obtained for ELA.

select documents to show to users on the website. Participants were free to decide the number of labels, as well the criteria to include a label in the list. They were also free to decide when to stop and which criteria to use to evaluate the quality of the rules used to specify to the label.

We prepared a dataset for the experiment by selecting a subset of the documents from the *Yelp Open Dataset* [39]. This dataset contains a set of reviews written by users for different types of businesses. Each review contains, among other information, the *text of the review*, the *business name* and the *type of the business*, where each business may be related to one or more types. From this dataset, we selected the top 9 most relevant, but also distinct, business types, and considered only documents associated with these business types. To control for the different number of reviews for each type, we used stratified sampling to obtain a more uniform distribution of reviews.This resulted in a dataset with $19,348$ reviews.

We recruited 9 participants, all with prior experience with computers and data analysis. Each participant received 15 minutes of training on how to use the tool and was allowed to ask questions during this session. The participants then received the description of the task and had 1 hour to create their solution for the task, i.e., a set of labels and rules. Participants were instructed to speak while creating their solution following a "think aloud" protocol. While the participants performed the task, an investigator took notes related to the actions performed. We also recorded the participants' screen and logs to enable a more fine grained analysis of interactions. After completion, the participants filled a questionnaire to express their impressions on the usability of the system. We also conducted semi-structured interviews to gather subjective feedback and to ask clarifications about interaction patterns and analysis strategies observed during the test. Participants received $30 *US Dollars* for their participation.

Usability

Given the level of complexity of our system and of the assigned task, achieving a good usability score is particularly important. We evaluated the usability of the system using the System Usability Scale [4]. This scale has proven to be a valuable and robust tool for assessing the quality of system interfaces [1].

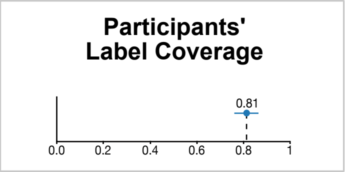


Figure 5. Bootstraped 95% confidence interval of number of labels covered by the participants' specification.

To compute the SUS score for the tool, at the end of the experiment we asked participants to fill out a survey with the 10 questions specified by the SUS test, grading them on a scale from 1 to 5, where 1 means strongly disagree and 5 means strongly agree. The SUS score grades systems on a scale between 1 and 100 and our system obtained an average score of 85.27 with 95% CI [80.72, 89.82]. According to Bangor et al. [1] a SUS score above 85 is considered excellent and is in the 4th top percentile range. Therefore the SUS score obtained for the tool was satisfactory.

We also asked participants about their satisfaction with the results and 7 out of 9 reported being satisfied or very satisfied while the other participants reported being neither satisfied nor unsatisfied. Most participants also were confident that they covered most of the labels in the collection, with 6 participants reporting being confident or very confident, and only 1 reporting being not confident. When asked if they believed the specifications they provided would correctly match relevant documents, 7 were confident or very confident, and 2 were neither confident nor confident.

Again, given the complexity of the task and the user interface we believe these results demonstrate that users could perform the exploratory labeling task with a high degree of confidence and achieve results they are sufficiently satisfied with.

Expected and unexpected patterns

Following the visual analytics mantra of "detecting the expected and discovering the unexpected" [38], a goal of label ideation phase in the exploratory labeling task, is to help users discover new labels in the data, i.e, help users update their mental model in the presence of the data, allowing them to find labels they did not expect. To measure how the labeling task impacts the mental model of a user, we asked the participants to guess which label they would be able to find before exploring the data and then compared their guesses with the final set of labels they generated. At the end of our study, 6 participants believed their final solution was different or very different from what they expected, 2 believed the solution was somewhat similar and only 1 participant believed the solution produced was similar to his prior expectations.

Coverage

A relevant goal of exploratory labeling is to achieve high coverage of the identifiable label. To test the performance of our method in this respect, we measure how many of the selected set of categories found in the Yelp data set our participants were able to discover. In order to develop a robust matching between the labels generated by the participants and the original labels we crowdsourced the matching task to a group of workers in Amazon Mechanical Turk. The workers were

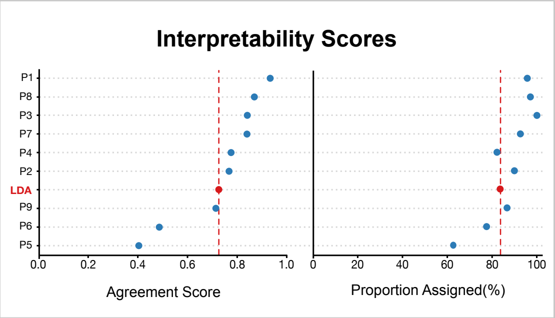


Figure 6. Agreement and assignment scores for each participants and LDA ordered by assignment score.

presented with one label at a time, from those generated by our participants, and had to assign one of the Yelp categories or specify that no matching was possible.

Each label from each participant was labeled by 5 different raters. Figure 5 shows the results of this analysis. The overall average coverage across the 9 participants was of 81%. When we look at coverage scores of individual participants, we can see there is considerable variability. While some participants were able to achieve a high coverage, a few showed a somewhat low coverage. This divergence can, at least partially, be explained by a pattern we observed during the test. Some of the participants decided to follow a label generation strategy that would prioritize label specification over discovery. As a consequence, given the fixed amount of time we gave to our participants, some of them could not adequately explore all possible labels they wanted to explore.

Specification Interpretability

While during exploratory labeling users provide names to the labels created, we expected the terms used to create the rules in the specification to provide confirmation of those labels, i.e., by looking a the terms used, one should be able to guess what the title of the label should be. We call this interpretability.

To measure interpretability, we use raters agreement as a proxy. If raters agree most of the time on which label matches which specification, we consider the specification highly interpretable. We compute the agreement using the *Fleiss score* [20] (a generalization of the Kappa score). We slightly modify the score to consider the cases in which the raters could not find a match between the label and the category as disagreements. In addition to the agreement, we also compute an *assignment score* that represents the proportion of labels that were assigned to a ground truth label. Figure 6 shows the results for the *agreement* and *assignment* scores.

In order to ground the scores on some relevant benchmark, we compare the results of the scores obtained by using LDA to those generated by our participants (Figure 6). The average *agreement score* of our participants and LDA are similar, 0.74 and 0.72 respectively. Similarly, for the *assignment* score, the average for the participants and LDA are 0.88 and 0.83 respectively. Similarly to the results we observed in the coverage analysis, we can see that these scores vary considerably across the study participants, with the majority of participants performing significantly better than LDA and a few performing significantly worse.

Automated Labeling

An important aspect of the exploratory labeling process is that at the end of the process labels can, at least in principle, be used to create classifiers that automatically classify the existing data, as well as unseen data, into the set of labels defined during the process. Even if the emphasis of this work is on generating labels and their specifications, in this section we want to more closely explore the existing options when one wants to use the result of exploratory labeling to perform classification.

To this purpose, we propose four different strategies. In the first strategy, we use the document matching function we defined above to decide which labels correspond to each document. In the second strategy, we use the same document matching function with an additional threshold, which removes all the documents that score below the first percentile of the score distribution. In the third strategy, we train an *SVM* classifier using the documents retrieved by the first strategy as training set. Finally, in the fourth strategy, we also train an *SVM* classifier but we use the documents retrieved by the second strategy, that is, the one using a threshold.

The main idea behind testing these four strategies is that we want to see if using a classifier we can achieve more generalizable results. We also want to see if applying a cutoff has any measurable impact on performance.

Since different users tend to produce different results we first had to decide which labeling results to use for this analysis. Furthermore, since the quality of specifications built during the study arguably influences the quality of the results, we had to find a way to generate a specification "gold standard". To this purpose, we randomly selected 3 of the participants to perform an additional label specification task. In this task we provided the participants with the right set of categories extracted from the *Yelp* dataset and asked them to create the best possible specifications.

In order to compare the performance of the four strategies we split the data into training and test data and use *precision* and *recall* scores for evaluation. The scores are averaged across the results obtained with the data coming from the three participants.

Figure 7 shows the mean and the 95% bootstrapped confidence intervals for *precision* and *recall* across labels and participants. We call the four strategies presented above as follows: *document matching*, *document matching + cutoff*, *SVM*, and *SVM + cutoff*. From the figure, we can see that all methods have a comparable level of precision, with a slight increase when using the SVM classifier. Where we observe a substantial improvement is in the recall score, which is substantially higher for the SVM classifier, which confirms our intuition that using a classifier can provide more generalizable results. We also notice some improvement with the cutoff version of the methods but not as significant as expected.

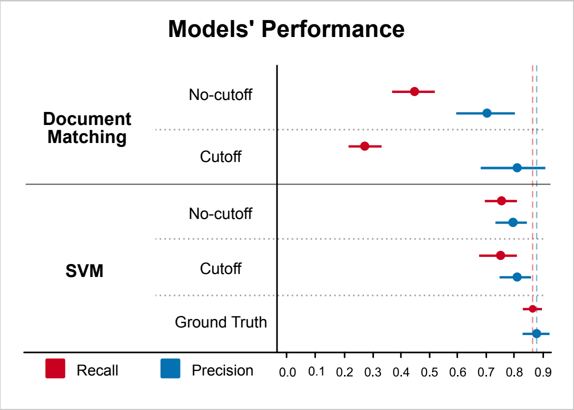


Figure 7. Comparison between the performance of different models, 1) created by the users and 2) SVM classifiers trained over the participant's output.

In light of these results, we suggest the use of our proposed fourth strategy when at the end of the exploratory labeling process one wants to use the labels to generate effective classifiers. One question that remains to verify is whether using this strategy one may be able to build a classifier with performance comparable to that of a classifier trained on the original data which contains the actual ground truth.

To clarify, the four strategies we just discussed are all based on labels generated through the exploratory labeling process, not those originally found in the data. For this reason we deem interesting performing one last comparison with an SVM model built on the original data. As shown in Figure 7, the performance of the model built on ground truth outperforms all the others. The results are however encouraging considering the unsupervised nature of these strategies.

DISCUSSION

In this work, we presented exploratory labeling as a highly interactive human-driven activity to generate labels and specifications out of a document collection. We also presented a system to perform exploratory labeling with the support of a series of computational methods aimed at guiding the user in the process. The most important questions we had for this work were about the usability of our approach and the quality of the results generated.

From our experiments, the usability of the system seems to be satisfactory. The task is clearly a complex one and requires extended time and sustained effort. However, our participants were able to produce reasonable results with very little training; having performed this type of task for the first time.

An important observation, looking at the results obtained from the *coverage* and *interpretability* tests is that the performance of the participants varied substantially. A minority of participants achieved low scores in both our metrics and a substantial number of participants achieved scores equal or better than the output generated by an automated model. We do not know precisely what the source of this variability is. It may be individual differences among the participants or, as we mentioned before, it may reflect different strategies the participants decided to use at the beginning and that led to sub-optimal results. In this second case, proper training before embarking on an exploratory labeling session may of help.

A somewhat surprising result is the one we obtained with interpretability. Our initial hypothesis was that results generated by topic modeling (LDA) would be way less interpretable than those generated by our human-driven method. The results of our experiment, however, do not confirm the existence of such a large difference. There are many possible reasons behind these results. One is that the data set and task we used did not generate a particularly problematic output from topic modeling. Another explanation, however, is that human perception mechanisms are robust to noise and for this reason, it was not substantially harder to match topics with labels when using the output of LDA. More research in this direction is needed to shed light on this problem.

In the section about automated labeling, we demonstrated that using a classifier can greatly enhance the performance of the method if one wants to use the labeling output to build an actual classifier. Even more important is our finding that building a classifier using the output of exploratory labeling generates results that are comparable to those obtained by a classifier trained on the original ground truth data. This is encouraging because it suggests exploratory labeling can help people uncover meaningful *unknown* labels from the corpus.

One open issue we want to investigate in the future is how our proposed approach compares to approaches that rely more heavily on existing unsupervised learning methods. More precisely, our method promotes a solution to the problem that requires a deep involvement of the user in generating the labels. An alternative approach may be one where unsupervised methods create a good-enough solution first and then the user is able to modify and explore such solution to adapt it to the needs of the user. More research is needed to better understand which paradigm works best under which condition.

CONCLUSION AND FUTURE WORK

In this paper, we defined *Exploratory Labeling*, a task that aims to ideate and specify labels our of a document collection. We proposed a mixed-initiative method to support such task where the user is in control of the process, and it is aided by the machine that works as an assistant providing recommendations and helping the user to evaluate the results of the task. We validate our method through an application in a real-world scenario and a series of experiments conducted with the goal of understanding the extent to which the method works and its limitations. This work makes a few first steps in the direction of interactive solutions that are able to take advantage of the machine computational power but also the user's domain knowledge. This work describes some initial findings, but more research is needed. Specifically, we plan to investigate the effect of different levels of recommendations: how starting with a blank canvas and building the solution based on recommendations differ from the system creating an initial solution leaving to the user only the job of edit and improve it.

ACKNOWLEDGMENTS

This work is supported in part by CAPES Foundation, Ministry of Education of Brazil - process: BEX 13235/13-3.

REFERENCES

1. Aaron Bangor, Philip T Kortum, and James T Miller. 2008. An empirical evaluation of the system usability scale. *International Journal of Human–Computer Interaction* 24, 6 (2008), 574–594.

2. Jürgen Bernard, Marco Hutter, Matthias Zeppelzauer, Dieter Fellner, and Michael Sedlmair. 2018a. Comparing Visual-Interactive Labeling with Active Learning: An Experimental Study. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 298–308.

3. Jürgen Bernard, Matthias Zeppelzauer, Michael Sedlmair, and Wolfgang Aigner. 2018b. VIAL: a unified process for visual interactive labeling. *The Visual Computer* (2018), 1–19.

4. John Brooke and others. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.

5. Davide Ceneda, Theresia Gschwandtner, Thorsten May, Silvia Miksch, Hans-Jörg Schulz, Marc Streit, and Christian Tominski. 2017. Characterizing guidance in visual analytics. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 111–120.

6. Senthil Chandrasegaran, Sriram Karthik Badam, Lorraine Kisselburgh, Karthik Ramani, and Niklas Elmqvist. 2017. Integrating visual analytics support for grounded theory practice in qualitative text analysis. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 201–212.

7. Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L Boyd-Graber, and David M Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*. 288–296.

8. Kathy Charmaz. 2014. *Constructing grounded theory*. Sage.

9. Nan-Chen Chen, Margaret Drouhard, Rafal Kocielnik, Jina Suh, and Cecilia R. Aragon. 2018. Using Machine Learning to Support Qualitative Coding in Social Science: Shifting the Focus to Ambiguity. *ACM Trans. Interact. Intell. Syst.* 8, 2 (2018), 9:1–9:20.

10. Jaegul Choo, Changhyun Lee, Hannah Kim, Hanseung Lee, Zhicheng Liu, Ramakrishnan Kannan, Charles D Stolper, John Stasko, Barry L Drake, and Haesun Park. 2014. VisIRR: Visual analytics for information retrieval and recommendation with large-scale document data. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 243–244.

11. Jaegul Choo, Changhyun Lee, Chandan K Reddy, and Haesun Park. 2013. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 1992–2001.

12. Rudi L Cilibrasi and Paul MB Vitanyi. 2007. The Google Similarity Distance. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 19, 3 (2007), 370–383.

13. Kristin Cook, Nick Cramer, David Israel, Michael Wolverton, Joe Bruce, Russ Burtner, and Alex Endert. 2015. Mixed-initiative visual analytics using task-driven recommendations. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 9–16.

14. Wenwen Dou, Xiaoyu Wang, Drew Skau, William Ribarsky, and Michelle X Zhou. 2012. Leadline: Interactive visual analysis of text data through event identification and exploration. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 93–102.

15. Wenwen Dou, Li Yu, Xiaoyu Wang, Zhiqiang Ma, and William Ribarsky. 2013. Hierarchicaltopics: Visually exploring large text collections using topic hierarchies. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2002–2011.

16. Margaret Drouhard, Nan-Chen Chen, Jina Suh, Rafal Kocielnik, Vanessa Pena-Araya, Keting Cen, Xiangyi Zheng, and Cecilia R Aragon. 2017. Aeonium: Visual analytics to support collaborative qualitative coding. In *IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 220–229.

17. Steven M Drucker, Danyel Fisher, and Sumit Basu. 2011. Helping users sort faster with adaptive machine learning recommendations. In *IFIP Conference on Human-Computer Interaction*. Springer, 187–203.

18. Mennatallah El-Assady, Rita Sevastjanova, Fabian Sperrle, Daniel Keim, and Christopher Collins. 2018. Progressive Learning of Topic Modeling Parameters: A Visual Analytics Framework. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 382–391.

19. C. Felix, S. Franconeri, and E. Bertini. 2018. Taking Word Clouds Apart: An Empirical Investigation of the Design Space for Keyword Summaries. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 657–666.

20. Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin* 76, 5 (1971), 378.

21. Benjamin Höferlin, Rudolf Netzel, Markus Höferlin, Daniel Weiskopf, and Gunther Heidemann. 2012. Inter-active learning of ad-hoc classifiers for video visual analytics. In *, 2012 IEEE Symposium on Visual Analytics Science and Technology (VAST)*. IEEE, 23–32.

22. Matthew Hoffman, Francis R Bach, and David M Blei. 2010. Online learning for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*. 856–864.

23. Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 159–166.

24. Minjeong Kim, Kyeongpil Kang, Deokgun Park, Jaegul Choo, and Niklas Elmqvist. 2017. Topiclens: Efficient multi-level visual topic exploration of large-scale document collections. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 151–160.

25. Todd Kulesza, Saleema Amershi, Rich Caruana, Danyel Fisher, and Denis Charles. 2014. Structured labeling for facilitating concept evolution in machine learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3075–3084.

26. Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*. 1188–1196.

27. Hanseung Lee, Jaeyeon Kihm, Jaegul Choo, John Stasko, and Haesun Park. 2012. iVisClustering: An interactive visual document clustering via topic modeling. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 1155–1164.

28. Tak Yeon Lee, Alison Smith, Kevin Seppi, Niklas Elmqvist, Jordan Boyd-Graber, and Leah Findlater. 2017. The human touch: How non-expert users perceive, interpret, and fix topic models. *International Journal of Human-Computer Studies* 105 (2017), 28–42.

29. Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

30. Megh Marathe and Kentaro Toyama. 2018. Semi-Automated Coding for Qualitative Research: A User-Centered Inquiry and Initial Prototypes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, 348:1–348:12.

31. Fredrik Olsson. 2009. A literature survey of active machine learning in the context of natural language processing. (2009).

32. Deokgun Park, Seungyeon Kim, Jurim Lee, Jaegul Choo, Nicholas Diakopoulos, and Niklas Elmqvist. 2018. ConceptVector: text visual analytics via interactive lexicon building using word embedding. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 361–370.

33. Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543. `http://www.aclweb.org/anthology/D14-1162`

34. Daniel Ramage, Evan Rosen, Jason Chuang, Christopher D. Manning, and Daniel A. McFarland. 2009. Topic Modeling for the Social Sciences. In *Workshop on Applications for Topic Models, NIPS*. `http://vis.stanford.edu/papers/topic-modeling-social-sciences`

35. Christin Seifert and Michael Granitzer. 2010. User-based active learning. In *IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 418–425.

36. Burr Settles. 2012. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6, 1 (2012), 1–114.

37. Ehsan Sherkat, Seyednaser Nourashrafeddin, Evangelos E Milios, and Rosane Minghim. 2018. Interactive Document Clustering Revisited: A Visual Analytics Approach. In *23rd International Conference on Intelligent User Interfaces*. ACM, 281–292.

38. James J Thomas and Kristin A Cook. 2006. A visual analytics agenda. *IEEE computer graphics and applications* 26, 1 (2006), 10–13.

39. Yelp. 2018. Yelp Open Dataset. (2018). Retrieved March 01, 2018 from `https://www.yelp.com/dataset`.