ExNav: An Interactive Big Data Exploration Framework for Big Unstructured Data

Xiaoyu Ge, Xiaozhong Zhang and Panos K. Chrysanthis
Department of Computer Science, University of Pittsburgh
{xiaoyu, xiaozhong, panos}@cs.pitt.edu

2020 IEEE International Conference on Big Data (Big Data) | 978-1-7281-6251-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/BigData50022.2020.9377741

***Abstract*—Driven by the increasing gap between the exponential growth of data and the limited human ability to comprehend them, recently, a novel interactive data exploration approach called Explore-by-Examples has generated a lot of attention for its capabilities to bridge this gap and to help the user obtain high-value content from the data that are often hidden using the traditional search methods. However, despite their effectiveness in extracting valuable information, existing Explore-by-Examples systems focus solely on structured data, which represents a small portion of the data available today. In this work, we present a novel data exploration framework, namely ExNav (Exploration Navigator), which enables the user to effortlessly explore the world of unstructured data for insights that are often unreachable from traditional search and exploration methods. In particular, we exploit the space of advanced machine learning, data embedding, and active learning algorithms to design effective exploration and space pruning approaches tailored for unstructured datasets. Our experimental evaluation using multiple real-world unstructured datasets (i.e., text, image, and graph) show that ExNav can reduce users' effort by up to 9x while still achieving the same accuracy as the state-of-the-art alternative. Moreover, ExNav is also able to identify relevant data items that are often undetectable by current techniques, even when a large number of samples are explored.**

I. Introduction

Search is central to our daily interactions with data, ranging from rudimentary tasks such as online shopping to more complex ones such as retail analytics. With the rapid growth of the complexity and volume of available data, the traditional search methods relying on explicit keywords or queries can quickly lose their effectiveness. As reported in recent studies [1], it is often difficult for users to construct precise articulations that describe their interests. In such cases, traditional search methods usually fail to deliver satisfying results, and the user often needs to deal with results that are too big in size due to loose queries or keywords. Consequently, to obtain a satisfying result, users need to execute numerous ad-hoc queries with tightened conditionals to reduce the search space, which requires a considerable amount of time and human effort.

In order to address the challenges outlined above, recent research efforts have been directed towards designing data exploration techniques that aim to assist users in finding their intended items (e.g., [2], [3], [4], [5], [6], [7], [8]). Examples of such techniques include *Query Recommendation* (e.g., [2], [3]), *Query Refinement* (e.g., [4], [5]), and *Explore-by-Examples* (e.g., [6], [9], [10], [11]). Among those techniques, Explore-by-Examples is rapidly becoming an attractive choice, due to its efficiency in finding relevant items that are often undiscoverable using traditional search methods. In particular, Explore-by-Examples does not require users to formulate any complex queries, nor does it need any form of description of the target items. The entire exploration can be done by answering simple binary (i.e., yes or no) questions. This is as opposed to Query Recommendation techniques that require intensive query logs and user profiles, which are often unavailable when users are exploring datasets for the first time. Similarly, Query Refinement techniques require the user to provide some initial imprecise queries to be progressively refined into a more precise one, a process that clearly requires users to possess some good understanding of the underlying database schema as well as the ability to formulate meaningful queries. More importantly, Explore-by-Examples based exploration systems can further be used to enhance traditional search results. Especially, they can be used to address the problem of having overwhelming results due to a broad query or search conditions [6], [10].

To further explain how Explore-by-Examples works, consider the scenario in which a user is searching for "interesting" articles or tweets regarding COVID-19. The user has a mental representation of the characteristics of the COVID-19 articles that suit his/her interest but he/she are not fully acquainted with the actual keywords, author, or topic that would select these interesting articles. Given the fact that a simple keyword of "COVID-19" leads to a deluge of COVID-19 news and articles, mining all or most relevant articles, news, and tweets that satisfy the user's specific interest becomes another "mission impossible". However, if we show the same user an example article or news title that they have seen and ask them "Is this the type of article that you have in mind? Yes or no?", people can answer such ordinal questions with more ease than absolute judgments (i.e., finding the exact words or queries to describe their interest) [12].

The aforementioned scenario is not restricted to searching for articles or news only. Browsing for images, videos, or searching for nodes in large interconnected graphs, users face the same challenge of not being able to accurately describe items that are of interest using traditional search interfaces. However, a user can easily provide a relative judgment based on an actual example, that is, answering questions such as "Do you find this image example interesting: Yes or No?", rather than describing in detail the characteristics of their intended images.

Motivated by the above scenarios, the main idea underlying the Explore-by-Examples techniques, which is the focus of this work, is to automatically discover all relevant items intended by the user based on their feedback on a small set of sample objects, called "examples" [6], [10], [11]. Typically, an Explore-by-Examples exploration starts with a large set of initial data resulting from a loose search query that contains mostly uninteresting data items with a small percentage of interesting items. Later, the user is prompted

to label a small set of strategically collected examples as *relevant* or *irrelevant* to her exploration task. Based on the feedback, the system generates a predictive model that is used to collect a new set of sample objects. Subsequently, these new samples are presented to the user, and her relevance feedback is incorporated into the model. In the background, the system leverages the predictive model to discover and identify more objects relevant to the user's task. A list of potentially relevant objects can be retrieved at any time during the exploration, and the exploration stops once the user is satisfied with the results.

Despite being effective in retrieving relevant data items, existing Explore-by-Examples systems focus only on structured data with low dimensionality, which does not apply to unstructured data (e.g., image, text, and graph) due to their more complex nature and much higher dimensionality. In this paper, we describe our framework, coined *ExNav* (**Ex**ploration **Nav**igator), which is the first interactive data exploration framework that leverages Explore-by-Examples paradigm for unstructured data. ExNav is capable of supporting any form of unstructured data as long as an underlying vector representation of the data can be created. It dynamically refines the exploration space based on a set of interactions with the user, such that in each interaction, the user is presented with an example and is asked to provide only binary feedback (e.g., interesting or not interesting) on each example. Furthermore, it implements a number of optimizations that further improve its exploration efficiency and ensures the desired interactive performance during the exploration.

We experimentally evaluated ExNav on three real-world unstructured datasets (i.e., [13], [14], [15]), each with different data types. The experimental results show that ExNav can reduce users' effort by up to 92.3% while still achieving the same accuracy as the state-of-the-art alternative. Moreover, ExNav is also able to identify with high accuracy relevant data items that are often undetectable by current techniques, even when a large number of samples are explored.

Specifically, our contributions in this paper are as follows:

- We propose ExNav, an effective generic interactive data exploration framework that is tailored for unstructured data. Our ExNav is able to work with any unstructured data (e.g., images, text, audio, and graph) as long as an underlying embedding representation can be created (Sections II-III).
- We introduce several optimizations that reduce the time and effort required to learn the user's interest, and in turn, boost the accuracy of the result and reduce the number of examples required during the exploration (Section III).
- We realize several instances of ExNav framework for different data types and experimentally verify their effectiveness and efficiency with real-world image, text, and graph datasets. Our experimental results that ExNav significantly outperforms state-of-the-art data exploration systems in accuracy while achieving the necessary efficiency for interactive performance (Section IV).

II. Problem Definition & Background

In this section, we formally introduce our problem and provide the necessary background of our ExNav.

A. Data Exploration Task

To illustrate visually the data exploration task handled by ExNav, we assume each item of the data has been reduced to 2 dimensional. Figure 1, shows the data objects (e.g., image, article title) in 2-dimensional space, where each data object is represented by a 2-attribute data point. Meanwhile, the dashed rectangles represent regions of the data space that are of special interest to the user's current data exploration task. Consequently, the goal of Explore-by-Examples techniques, including ExNav, is to effectively identify all or most relevant items (items in dashed rectangles) with high precision.

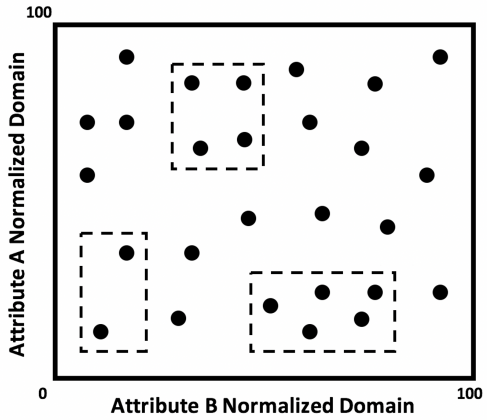

Fig. 1: 2-D data space with 3 relevant regions.

B. Problem Settings

The data exploration problem addressed in this work can be formalized as follows. Consider a d-dimensional data space D of size N data items created using an embedding algorithm. Further, consider L, a small subset of D of size n data items, where $n << N$. Each data item in L is interactively labeled by the user as relevant or irrelevant. The goal is to construct a predictive model ρ, which accurately predicts the set of regions of the data space $\Re^+$ that are of interest to the user using the labeled data items in L. That is, all data items predicted by ρ as positive are relevant to user exploration, and all data items predicted by ρ as negative are irrelevant.

Given the fact that the user is unaware of the proper specification that describes $\Re^+$, the user can only recognize it in hindsight based on the data items predicted as relevant and retrieved by ρ. Hence, the objective is to predict accurately all regions in $\Re^+$, which can be naturally measured using the *F-measurement* [6], [10], the harmonic mean between precision and recall. Particularly, for q data space D of size N and a predictive model ρ, all the data items in D predicted by ρ as positive should be relevant to user exploration, and the remaining data items in D should be irrelevant. Accordingly, our goal is to design a solution that would maximize the F-measure for a fixed amount of user labels, such that F-measure is defined as:

$$F(N) = \frac{2 \cdot \text{Precision}(N) \cdot \text{Recall}(N)}{\text{Precision}(N) + \text{Recall}(N)} \quad (1)$$

Here, precision measures the portion of true relevant data items among all the data items predicted as relevant by ρ. Hence, true relevant, or true positive, indicates that a data item is both relevant to the user and has been predicted as relevant by ρ. If a data item is irrelevant but predicted to be relevant by ρ, it is considered a false positive. Recall measures the ratio

of the true relevant data items captured by ρ to all the data items that are actually relevant to the user.

C. Active Learning

As the goal of the predictive model in Explore-by-Examples is to seek the most informative next example to be labeled from a large dataset, it is aligned with the goal of *Active Learning* [16]. Active learning refers to a set of approaches that aim to learn an accuracy model with minimum labeled data for regression and classification tasks. One key component of active learning is the *query strategy* that sequentially selects the most informative unlabeled sample (i.e., data object) from the entire database to be labeled by the user.

In previous literatures, numerous query strategies [16] have been proposed to define the "informativeness" of samples, including: *Uncertainty Sampling*, *Query-By-Committee*, *Expected Model Change*, *Expected Error Reduction*, and *Expected Model Output Change*. Among these query strategies, Uncertainty Sampling is the most commonly used one because of its simplicity and efficiency, as pointed out in [16].

1) Uncertainty Sampling: Uncertainty sampling [17] is a query strategy that can be used with any probability-based classification model (e.g., Naive Bayes, SVM, etc..). The intuition underlying uncertainty sampling is that patterns with high uncertainty are hard to classify, and thus, if the labels of those patterns are obtained, they can boost the accuracy of the classification models. Particularly, in binary classification models (e.g., with class labels 0 and 1), the most uncertain example $\mathbf{x}$ is the one which can be assigned to either class label $z(\mathbf{x})$ with probability 0.5. Inspired by the idea of uncertainty, also known as *least confidence*, [17] proposes a measurement of uncertainty for binary classification models:

$$u^{(lc)}(\mathbf{x}) = 1 - p(\hat{y}|\mathbf{x}) \tag{2}$$

where $u^{(lc)}(\mathbf{x})$ is the uncertainty score with the least confidence measurement of $\mathbf{x}$, and $\hat{y}$ is the predicted class label of the unlabeled $\mathbf{x}$. Accordingly, after measuring the uncertainty of each unlabeled sample, the unlabeled sample with highest uncertainty is selected:

$$\mathbf{x}^* = \text{argmax}_{\mathbf{x}} u(\mathbf{x}) \tag{3}$$

where $u(\mathbf{x})$ can be any other measurement of informativeness over the unlabeled sample $\mathbf{x}$.

III. The Exploration Navigator (ExNav)

In this section, we formally describe our proposed ExNav.

A. Proposed Solution

Our solution ExNav is designed to be independent of the underlying types of data and aims at the following goals 1) minimize the user labeling effort during an exploration task, and 2) obtain all data objects relevant to user exploration with the highest accuracy.

As shown in Algorithm 1, ExNav identifies relevant objects through an iterative interaction with the user. In each iteration, ExNav presents the user with one *example* (e.g., an image, a news title) from the entire dataset and requests the user's feedback on the relevance of this example to his/her exploration. Inside the system, ExNav decides and utilizes the *data embedding* method based on the data type involved in the exploration to cover each data item into its corresponding feature representations (i.e., embeddings) (Line 1). Leveraging these embeddings, ExNav employs a *predictive model* to wisely select the example to be presented that will maximize the benefit to the exploration task once its relevance with respect to the exploration is provided by the user (Lines 5-7). Once labeled, this example, along with its label (i.e., user feedback) will be incorporated with all previously labeled examples to update the predictive model for better subsequent explorations (Lines 8-11). The user terminates the exploration once he/she is satisfied with the exploration results, and the set of relevant data objects captured by the most recent predictive model will be returned (Line 13).

Algorithm 1 The Exploration Navigator

Require: The raw data set D
Ensure: The set of relevant objects R
1: Convert D into a set of embeddings E
2: $E_s \leftarrow dataReduction(E)$
3: $L \leftarrow$ obtain initial set of samples
4: Unlabeled set $U \leftarrow E_s$
5: $M \leftarrow$ initialize query selection method
6: **while** user continues the exploration **do**
7: Choose one x from U using M
8: Solicit user's label on x
9: $L \leftarrow L \cup \{x\}$
10: $U \leftarrow U - \{x\}$
11: $M \leftarrow$ trained with L to update M
12: **end while**
13: Return relevant objects R captured by the most recent M

In the following sections, we will present each main component of ExNav in detail.

B. Data Embedding

Currently, there exists a large body of embedding algorithms that encompasses various methods for learning of feature representations of different unstructured data types (e.g., image, graph, text) in numerical vector space. Typically, embedding methods aim to create embeddings (i.e., vector representations) based on the assumption that the similarity between data items in their original form should be reflected in the learned feature representations. To achieve this goal, a large variety of algorithms have been proposed for a wide range of data types using both supervised and unsupervised learning methods. For instance, Word2vec [18], Universal sentence encoder [19], and GloVe [20] are some well recognized methods for embedding text into its corresponding vector space. Similarly, for image data, there exists a large body of embedding methods that range from deep learning-based methods (e.g., Resnet-18 and Alexnet) to more traditional keypoint-based embedding methods (e.g., Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF)). Even for more complex data types such as graphs, the problem of embedding each data node into its vector space is also well studied (e.g., [21], [22]). These embedding methods provide good representations of its data items in the corresponding vector space that preserves the relative similarity between the data items in its original space. In our work, we leveraged these data embedding algorithms and designed ExNav to be a generic interactive data exploration framework that works with any unstructured data as long as an embedding representation can be produced.

Different embedding methods may produce embeddings with different dimensionality, which can range from 128 dimensions to 32768 dimensions [23]. Such a large variation in the length of the embeddings can lead to inconsistent

performance in the subsequent exploration task and slow convergence of the predictive model. Furthermore, many of these embeddings are loosely embedded, and thus, many of their attributes (i.e., dimensions) do not provide much value in identifying relevant data items. To address this challenge, in ExNav, we use dimensionality reduction algorithms to compress each high dimensional embedding into more condensed lower-dimensional vector representations. In particular, we used a well-known dimension reduction algorithm called *Principal Component Analysis* (PCA), which aims to minimize the reconstruction error between the compressed low dimensional vector and its original high dimensional representation.

C. Exploration Space Pruning

In a typical active learning setting, query strategies are applied over the entire set of unlabeled data objects to select the next example to be presented to the user for labeling. However, such exhaustive search incurs high processing costs, causing delays between each user-machine interactions, and leads to slower convergence of the predictive model. To address this challenge, data exploration approaches typically employ space pruning techniques for minimizing the number of unlabeled objects to be searched. In the ExNav framework, we consider simple yet effective space pruning methods, called *Multi-Instance Space pruning* (MIS pruning). Our MIS pruning is inspired by the Multi-Instance Active Learning (MIAL) [24], which is a set of popular approaches for reducing user labeling effort.

The idea in MIAL is that data objects can be grouped into a set of bags, and the user is asked to label each bag as positive or negative. Accordingly, MIAL assumes that a bag is negative if every object in that bag is negative; otherwise, the bag is positive. Subsequently, these bag-level labels will be used in the training of either bag-level or instance-level classifiers [25]. Compared to the more traditional instance-level active learning, MIAL has proven to be very effective in further reducing the labeling cost in many domains and applications [25]. For instance, in a text document, image, or a biological sequence, consider the case where individual instances tend to form very distinct clusters. In such cases, it may often be easier to label a group of strongly similar objects rather than each individual one, and all objects of the group inherit this label. Previous works have shown that MIAL efficiently reduces annotation costs in various applications such as object detection [26], web search results [27],and medical analysis [28], as well as different data types such as text [29], [30], image [28], and audio [31].

In contrast to MIAL that uses the bag-level labels for the training of classifiers, our MIS pruning uses the bag-level labels to identify and prune sub exploration spaces that are completely irrelevant to the exploration task, and in turn, reduces the amount of data that needs to be searched during the exploration. In particular, MIS pruning first uses the popular *K-Means* clustering algorithm to divide the items in the current exploration space into a set of bags based on their similarity. Given that each bag is created by grouping similar items, it ensures that the items within each bag are much more similar to each other compared to items across different bags. Thus, users only need to quickly skim through the items in each bag to see whether the bag is far away from the regions that contain relevant objects, without needing to closely examine each object of the bag. Such pruning technique enables us to leverage the benefit of MIAL and bag-level labels, while still preserving the high precision capability of instance-level active learning in identifying precisely all relevant regions.

D. Query Strategy

Query Strategy is the component of our ExNav framework that aims to minimize the labeling effort while maximizing the accuracy in discovering relevant objects. It is worth mentioning that in this context, a "query" refers to the process of selecting an example object to be presented to the user for labeling. In ExNav, we leverage uncertainty sampling to quickly learn the interest of the user and steer them towards all relevant data regions.

Uncertainty Sampling: As previously mentioned in Section II, uncertainty sampling is a popular active learning technique, which aims to choose the data points which are most beneficial to build a classification model that precisely distinguishes relevant and irrelevant data objects. According to Equation 2, to measure the uncertainty of a data object x, we need a probabilistic-based predictive model that would report the probability of x being positive or negative. Based on empirical studies, we selected a probabilistic based classification model, called *Gradient Boosting Decision Tree Classifier*, [32] to determine the uncertainty score.

Gradient Boosting Trees: Gradient boosting [33] is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models. Gradient boosting is typically used with decision trees, and the trees are trained in a sequential fashion, such that the subsequent tree is trained towards the residual (i.e., the difference between the observed value and the predicted value) of the current tree. For a binary classification problem, the observed value is 1.0 for positive (relevant) data points and 0.0 for the negative (irrelevant) data points. And the predicted value is the predicted relevance probability of the data point.

For example, to build the m^{th} tree, the algorithm would first calculate the residuals from the $(m-1)^{th}$ tree using:

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \ldots, n. \tag{4}$$

where $L(y_i, F(x_i))$ is the loss function and $F_{m-1}(x)$ is the output from the $(m-1)^{th}$ tree in log-odds form. Next, the algorithm would fit a new decision tree (i.e., the m^{th} tree) to the residuals r_{im}, which will have J_m terminal regions (i.e., leaves), $R_{1m}, \ldots, R_{J_m m}$. Then, the algorithm would calculate an output for each region using:

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma) \tag{5}$$

where x_i's are the data points, y_i is the observed value for x_i, and $F_{m-1}(x_i)$ is the predicted value for x_i in log-odds form from the $(m-1)^{th}$ tree.

Finally, the output of the m^{th} tree would be:

$$F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} \mathbf{1}_{R_{jm}}(x) \tag{6}$$

where ν is the learning rate to prevent over-fitting, and the summation is in place in case a data point ends up in multiple leaves.

Equation 6 is also used for the prediction of new data points, and the output is converted to probability using the sigmoid function. The predicted probability will, in turn, be used in the uncertainty calculation for the data point.

Uncertainty Sampling for Big Data: Uncertainty sampling is widely adopted in active learning to minimize the user labeling effort. However, traditional uncertainty sampling suffers from two major drawbacks: 1) *shortsightedness* (as point out in [34]), and 2) *low scalability* [10]. Shortsightedness refers to the issue where the uncertainty of unlabeled samples is estimated solely based on the information obtained from labeled samples, which usually represents only a tiny portion of the whole data space; therefore, it causes bias in the estimated uncertainty score, and in turn, leads to less effective decisions when selecting samples for labeling. Low scalability is caused by the fact that traditional uncertainty sampling requires performing an exhaustive search over the entire unlabeled datasets before one sample can be selected and presented to the user.

Randomized Uncertainty: To overcome the first drawback mentioned above, the work in [35] combines uncertainty with some degree of randomness. In particular, an unlabeled object that would be presented to the user as an example is probabilistically selected from the entire set of unlabeled objects. The probability that an unlabeled object x is selected is proportional to its uncertainty score:

$$p(x \text{ is selected}) = \frac{u(x)}{\sum_{x_u \in U} u(x_u)} \tag{7}$$

where U is the set of unlabeled objects and $u(\mathbf{x})$ is the uncertainty score of $\mathbf{x}$.

Since the probability that an unlabeled object x is chosen as an example is equal to its normalized uncertainty score, therefore, less uncertain objects can still have a small chance of being accepted as examples, which essentially reduces the bias introduced by the labeled samples.

Randomized Accept/Reject Uncertainty (RARU): When exploring large datasets, randomized uncertainty still suffers from low scalability due to the fact that the computation of the normalized uncertainty score requires going through all the unlabeled objects at least once. Thus, to address the second drawback of uncertainty sampling, we employed an accept/reject query strategy approach, called *Randomized Accept/Reject Uncertainty* (RARU) [10]. Particularly, in each iteration, RARU chooses the example to be presented to the user by randomly picking unlabeled objects from the entire set of unlabeled data, and then for each picked object x, RARU calculates the uncertainty score of x. Subsequently, RARU uses the uncertainty score of x as the way to determine its acceptance (i.e., whether to request user's feedback on x), such that the probability of an unlabeled data sample $\mathbf{x}$ being accepted under RARU is:

$$p(x \text{ is selected}) = \min_{k \in 0,1} \frac{Pr(C_k|\mathbf{x})}{0.5} \tag{8}$$

where $Pr(C_k|\mathbf{x})$ is the probability of $\mathbf{x}$ being assigned a binary class label C_k, and 0.5 is a normalizing factor since a prediction score of 0.5 indicates the classifier is most uncertain about an object.

RARU randomly visits each unlabeled object, until one object is accepted according to the above Equation 8, and the accepted object (i.e., example) will then be presented to the user for labeling. Clearly, RARU provides an early termination to the exhaustive search of the unlabeled data objects, while still preserving the feature of randomized uncertainty, which alleviates the issue of the shortsightedness of the traditional uncertainty sampling.

TABLE I: PARAMETERS

Number of data objects	6786 (All datasets)
Number of dimensions	32
Number of target relevent regions	1, 2, 3
Pruning question count	16
Example batch size	5
Max example allowed	3000
Target region cardinality	0.5% (S), 1.0% (M), 2.0% (L)
Considered Query Strategies	Traditional, RARU
Predictive Model	GBDT
Performance measure	F-Measure (Accuracy)
Schemes	RANDOM, REQUEST, ExNav_TU, ExNav_RARU
Number of runs per result	10

IV. Experimental Evaluation

In this section, we will present the results of our experiments. We begin this section by introducing the experimental setup. Then we demonstrate the performance of our schemes and of other alternatives.

A. Experiment Setup

Datasets In our experiments, we used three real-world unstructured datasets. The CBC news dataset (Text) contains news articles about COVID-19 [13]. The Caltech-256 image dataset (Image) contains images of 256 different categories [14]. The Mashup PPI dataset (Graph) contains nodes information in a protein-protein interaction graph [15].

Learning Representation We used off-the-shelf algorithms to generate the learning representations for the data. For the Text dataset, we used the Universal Sentence Encoder [19] to generate a 512 dimensional vector for each article. For the Image dataset, we used the ResNet [36] to generate a 512-dimensional vector for each image. For the Graph dataset, we used the pre-trained 500-dimensional vector from the Mashup project [37] for each node. Each vector is a compact representation of a node that accurately captures the topological patterns of the corresponding node in the original graph. All vectors in each dataset are projected to a 32-dimensional space using PCA as the final learning representation for that dataset and are used across all models.

Schemes We experimented with one baseline scheme, one state-of-the-art scheme, and two ExNav schemes. The baseline scheme is the random scheme (RANDOM), where the system selects examples randomly (based on uniform distribution) from the unlabeled set. The state-of-the-art scheme REQUEST [10] with all the recommended settings as described in the paper. The first scheme for ExNav is called ExNav_TU. It uses traditional uncertainty sampling [16] to select examples. The second scheme for ExNav is called ExNav_RARU. It uses RARU as the query strategy for example selection. We also use ExNav_TU_P and ExNav_RARU_P to denote ExNav_TU and ExNav_RARU with MIS pruning, respectively.

Target Interest Regions The exploration task characterizes user interests and eventually predicts the interest regions by iteratively gathering user labeled tuples. As mentioned before,

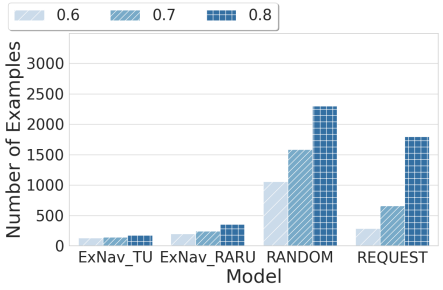

Fig. 2: Accuracy, 1 Large Region (Text Dataset)

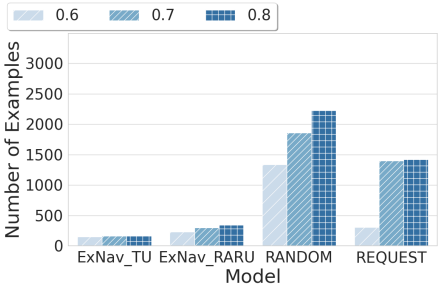

Fig. 3: Accuracy, 1 Large Region (Image Dataset)

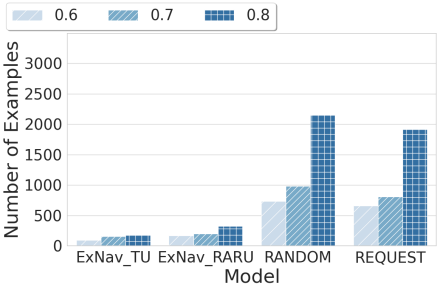

Fig. 4: Accuracy, 1 Large Region (Graph Dataset)

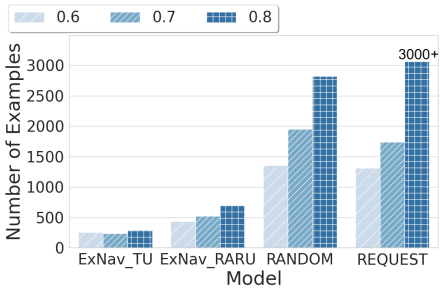

Fig. 5: Accuracy, 1 Small Region (Text Dataset)

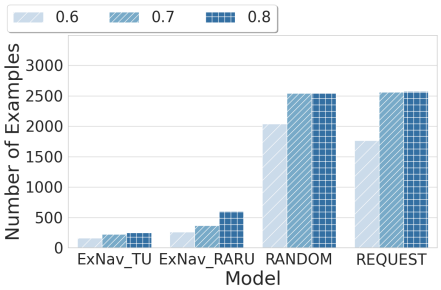

Fig. 6: Accuracy, 1 Small Region (Image Dataset)

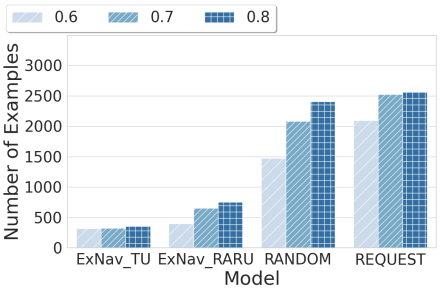

Fig. 7: Accuracy, 1 Small Region (Graph Dataset)

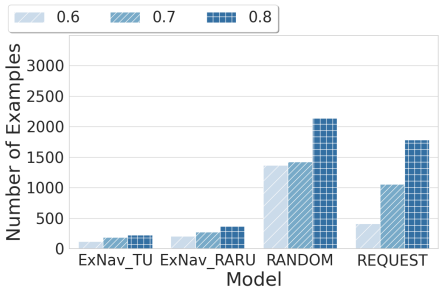

Fig. 8: Accuracy, 1 Medium Region (Text Dataset)

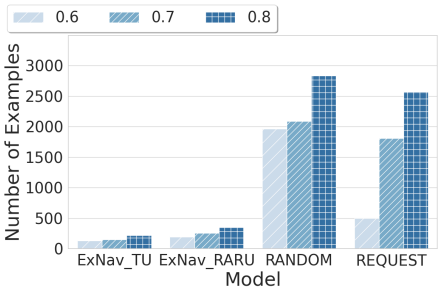

Fig. 9: Accuracy, 1 Medium Region (Image Dataset)

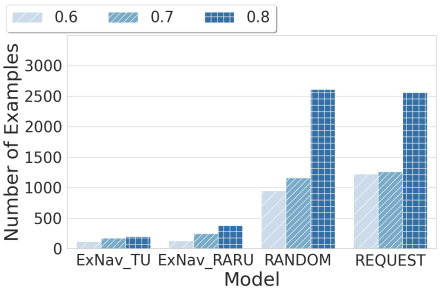

Fig. 10: Accuracy, 1 Mediums Region (Graph Dataset)

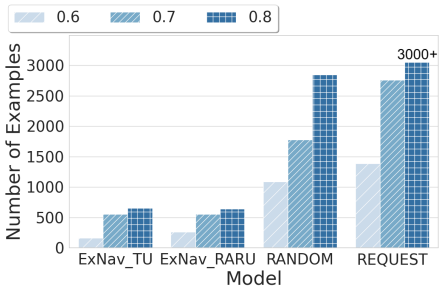

Fig. 11: Accuracy, 2 Medium Region (Text Dataset)

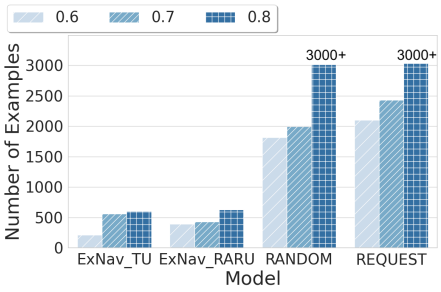

Fig. 12: Accuracy, 2 Medium Region (Image Dataset)

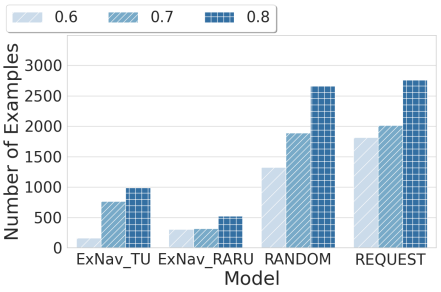

Fig. 13: Accuracy, 2 Mediums Region (Graph Dataset)

we focus on predicting the user interest regions. Particularly, in our experiments, we generate each target interest region (i.e., relevant region) with random range queries, and experimented with three different interest region amounts $\{1, 2, 3\}$. Further, we vary the single region complexity based on the data space coverage of the relevant regions. Specifically, we categorize relevant regions to small, medium, and large. Small regions have cardinality with an average of 0.5% of the entire exper-

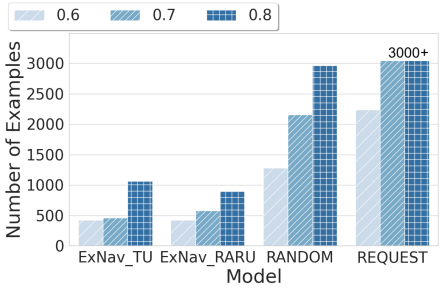

Fig. 14: Accuracy, 3 Medium Region (Text Dataset)

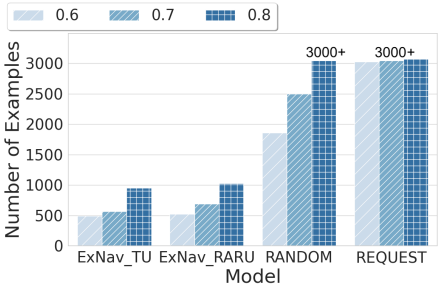

Fig. 15: Accuracy, 3 Medium Region (Image Dataset)

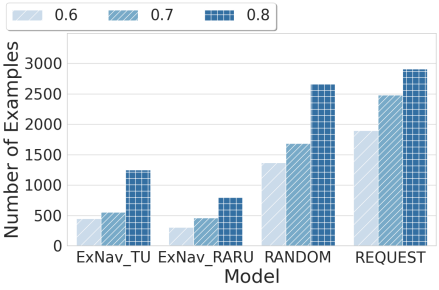

Fig. 16: Accuracy, 3 Mediums Region (Graph Dataset)

imental dataset, medium regions have a cardinality of 1.0%, and large regions have a cardinality of 2.0%.

User Simulation Given the target relevant regions, we simulate the user by collecting the exact target set of relevant tuples in the region. We rely on this "oracle" set to simulate user feedback for the multi-instance space pruning questions and the example relevance questions. For a pruning question, the user will give positive feedback if the bag covers at least one tuple in the "oracle" set, and a negative feedback if the bag does not cover any tuple in the "oracle" set. For a relevance question, the user will give positive feedback if the example is in the "oracle" set, and negative feedback is the example if not in the "oracle" set. This "oracle" set is also used as the ground truth set to evaluate the accuracy (F-measure) of our final predicted relevant regions.

Environment We implemented all algorithms with Python 3.8 and all the experiments were run on an Intel Core-i9 7980XE processor with 128GB RAM. All experiments reported are averages of 10 complete runs.

Parameters Table I shows a list of all settings and schemes of the experiment. In order to asses accurately the impact of different data types, by default, we extract 6786 distinct data objects for each data set, which is the size of the text data. Note using the same data object size across all three data sets allows us to show clearly the impact of different data types, as having inconsistent dataset size can influence the number of examples needed to achieve a certain accuracy level. In addition, the default batch size is 1. Note that the default size of the target relevant region cardinality is 1.0%, and default target relevant region size is 1 unless otherwise specified. The words "f-measure" and "accuracy" are used interchangeably in the text below.

B. Experimental Results

Accuracy Comparison Figures 2-16 show the number of examples needed to reach an accuracy (f-measure) of 60%, 70%, and 80% of all participating schemes with different target region numbers and three different datasets (i.e., text, image, and graph). Here we also vary the target region size from large to small. From these figures, we observe that both ExNav schemes have consistently demonstrated significantly higher effectiveness compared to REQUEST and RANDOM.

In particular, for the text data set, 1 relevant region, and ExNav_TU, for example, to reach an accuracy of 60%, ExNav_TU only requires around 125 examples for the large region, around 125 examples for the medium region, and around 250 examples for the small region on average. To achieve the same level of accuracy for the three target region sizes, REQUEST requires 2x, 3x, and 5x more examples, and RANDOM requires 8x, 11x, and 5x more examples, respectively. To reach an accuracy of 80%, ExNav_TU only requires around 180 examples for the large region, around 225 examples for the medium region, and around 325 examples for the small region on average. To achieve the same level of accuracy for the three target region sizes, REQUEST requires 10x, 8x, and >9x more examples, and RANDOM requires 13x, 9x, and 9x more examples, respectively. When we compare the performance for one relevant region, the largest deviation we see between ExNav and REQUEST is for image data set with one medium-sized relevant region, such that REQUEST requires 13x more examples than ExNav in order to reach an accuracy of 60%. Here, we would like to point out that 13x more examples required by REQUEST can be translated to a 92.3% reduction in users' effort when switching from REQUEST to ExNav. Note that in our experiment, we set the example limit to be 3000 as we believe it is not meaningful to label more examples beyond this point, we use the $>$ sign to indicate that the scheme is not able to reach the required accuracy level at 3000 examples.

Moreover, for different medium target region numbers, to reach an accuracy of 60%, ExNav_TU requires around 125 examples for 1 region, around 150 examples for 2 regions, and around 450 examples for 3 regions on average. To achieve the same level of accuracy for the three target region numbers, REQUEST requires 3x, 9x, and 5x more examples, and RANDOM would require 11x, 7x, and 3x more examples. To reach an accuracy of 80%, ExNav_TU only requires around 225 examples for 1 region, around 670 examples for 2 regions, and around 1050 examples for 3 regions on average. To achieve the same level of accuracy for the three target region numbers, REQUEST requires 8x, >4x, and >3x more examples, and RANDOM would require 9x, 4x, and 3x more examples, respectively. During the experiment, we also observed that one of the main reasons for REQUEST to perform poorly is due to its predictive model *Naive Bayes*, which is not able to overfit well with respect to these high dimension embeddings.

Furthermore, when looking at the two ExNav schemes, we observed that ExNav_TU on average performs better than ExNav_RARU in the case of only 1 relevant region. This is to be expected as ExNav_TU performs an exhaustive search over the entire exploration space in each iteration to find the most uncertain object for labeling. However, such benefit fades when more relevant regions exist in the space, such that we see ExNav_RARU on average outperforms ExNav_TU with 2

TABLE II: ExNav Runtime with Different Query Strategies

Query Strategy	10,000 instances runtime	100,000 instances runtime	1,000,000 instances runtime
Traditional Uncertainty	5.88 ms	61.02 ms	675.04 ms
RARU	0.51 ms	5.44 ms	52.79 ms

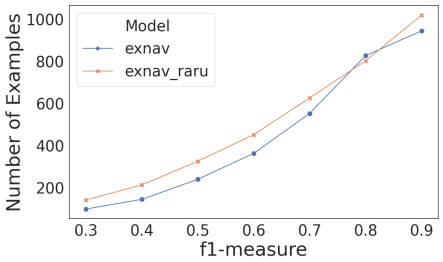

Fig. 17: Accuracy, 3 Medium Regions (Text Dataset)

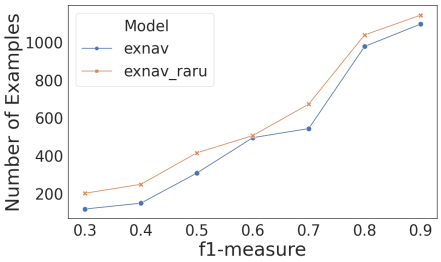

Fig. 18: Accuracy, 3 Medium Regions (Image Dataset)

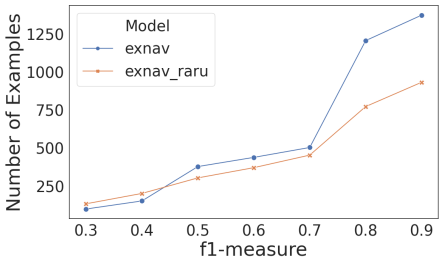

Fig. 19: Accuracy, 3 Medium Regions (Graph Dataset)

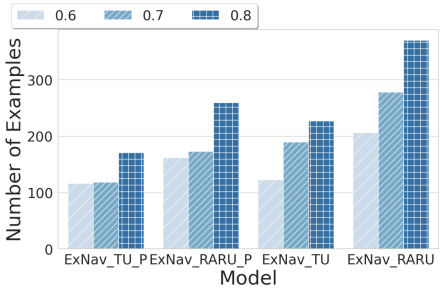

Fig. 20: Accuracy, 1 Medium Region (Text Dataset)

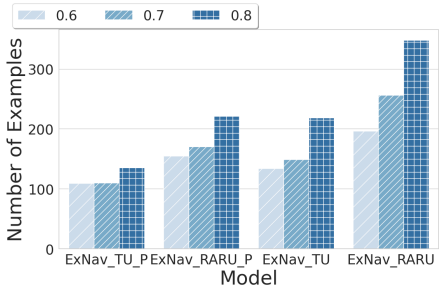

Fig. 21: Accuracy, 1 Medium Region (Image Dataset)

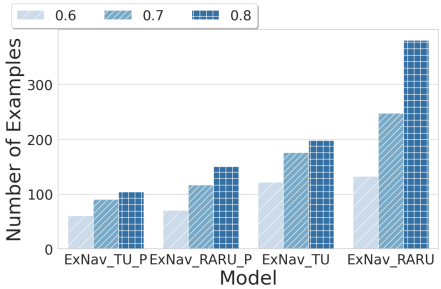

Fig. 22: Accuracy, 1 Medium Region (Graph Dataset)

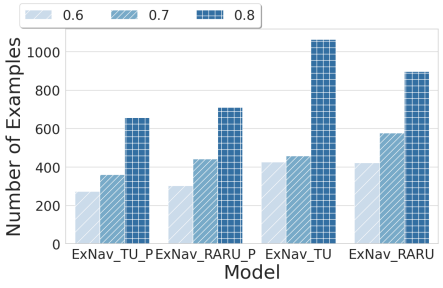

Fig. 23: Accuracy, 3 Medium Regions (Text Dataset)

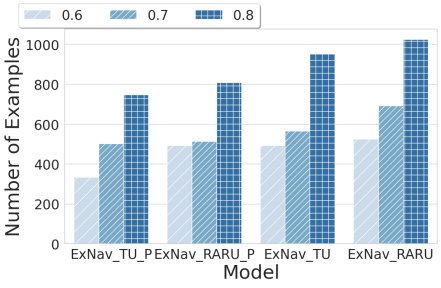

Fig. 24: Accuracy, 3 Medium Regions (Image Dataset)

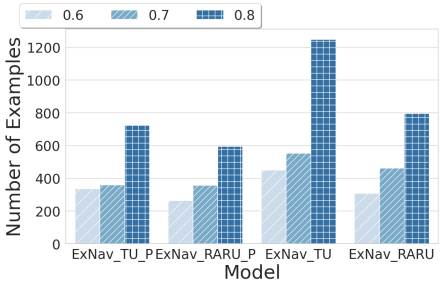

Fig. 25: Accuracy, 3 Medium Regions (Graph Dataset)

and 3 relevant regions by a noticeable margin. This is again as expected, since its limitation of shortsightedness, as discussed in Section III, will hinder its capability to discover multiple discrete relevant regions.

Lastly, as shown in the Figures 2-16, the results for the other two datasets also show similar trends for the accuracy comparison. To summarize, the two ExNav schemes have similar performance and consistently and significantly outperform REQUEST and RANDOM with respect to accuracy.

Scalability Table II, illustrates the runtime per iteration of ExNav under different query strategies. In particular, we extracted 10000 images from the Caltech-256 dataset and then duplicated the data in order to assess the runtime. The runtime for other data types is similar, as it is independent of a particular data type. The result showed that RARU helps to improve the scalability and the efficiency of the exploration by up to an order of magnitude. This is due to the fact that RARU does not require an exhaustive search over the entire exploration space, and thus, can deliver examples much quicker than traditional uncertainty sampling. These results, combined with the results of accuracy comparison, confirmed RARU's claim to address both the shortsightedness and low scalability drawback of the traditional uncertainty sampling. Furthermore, as Explore-by-Examples systems are often used as a post enhancement to the traditional keyword, faceted, or query search results, in these scenarios, the efficiency of ExNav can be further improved due to fewer data involved in the exploration.

Zoom-in into ExNav Schemes In order to better illustrate the comparison between our proposed schemes, in Figures 17-19, we zoom-in to the two ExNav schemes and show the number of examples needed to reach different levels of accuracy from 30% to 90%. For this set of comparisons, we consider a relatively complex scenario with three medium relevant regions. Compared to ExNav_TU, we have noticed that ExNav_RARU typically requires more labeled examples in the early stages of the experiment (e.g., below 70% of accuracy). This is due to the fact that in the early stages, to raise accuracy, it does not require all relevant regions to be identified accurately. However, when improving accuracy beyond 70%, ExNav_TU appears to be more struggled than ExNav_RARU (i.e., has a larger slope), which is expected due to its limitation of shortsightedness, as described in Section III.

Impact of Multi-instance Space Pruning As mentioned in the experiment setup, we have evaluated the effectiveness of our MIS pruning strategy. In particular, we created 16 bags by grouping similar objects and asking the user to tell us if any of the 16 bags are far away from the relevant region (i.e., contain no relevant objects). These regions will then be pruned from subsequent exploration. Note that the number of examples reported in the figures for ExNav_TU_P and ExNav_RARU_P does not include the 16 bags question, which has been applied as a fixed preprocess for these two schemes. The accuracy comparison between the ExNav schemes with and without pruning is shown in Figures 20-25. We can see that, for ExNav_TU_P, the pruning can reduce or save around 25%, 27%, and 40% user effort on average for the three datasets, respectively. For ExNav_RARU_P, the pruning can save around 26%, 21%, and 36% user effort on average for the three datasets, respectively. These results indicate that Multi-instance Space Pruning is effective in reducing the exploration space as well as the amount of feedback needed for exploration. Therefore, as discussed early in Section III, when exploring any data domains where group labels are inexpensive to provide, using ExNav with MIS Pruning is recommended.

V. Related Works

Interactive Data Exploration

As mentioned in [38], there are many works aiming to facilitate exploration for data analytics. Among the existing works, we review those directly relevant to us. Faceted search is a technique that iteratively suggests query attributes, which helps the user drill down into structured databases. However, it requires the user to either continuously provide attribute values until the desired set of relevant data objects are discovered [39], [40], [41], or provide certain quality measurements along with its threshold [42]. Semantic windows [43] is another data exploration technique that allows the user to interactively explore the data space with multidimensional shape-based and content-based predicates that are pre-defined. Unfortunately, the utility of this approach is restricted only to the case where such shape-based, or content-based patterns exactly match the user's interest. Both faceted search and semantic windows require the user to manually control the exploration direction, as opposed to our ExNav, where the system automatically steers the user towards all interesting tuples.

Two of the most recent works that are closely related to ours are AIDE [6] and REQUEST [10]. Both aim to interactively construct the exploratory query. The main idea underlying AIDE's predictive model for sample selection is to divide the space into equally sized subspaces and randomly sample one object from each subspace. Based on the user's feedback of this sample, AIDE would either sample more objects from its surrounding area, if the sample is relevant to the user's interest or further partition this subspace into smaller subspaces if the sample is irrelevant to the exploration. Recently, there is an improved version of AIDA [44], [11] that considered a new predictive model and space pruning technique for faster model convergence. However, this model can only discover one relevant region per each exploration session and is still optimized towards structured data, whereas our ExNav support multiple relevant regions and is designed for unstructured data.

REQUEST is the first Explore-by-Examples system that employs active learning for the selection of examples and divides the exploration process into two stages: *data reduction* and *query selection*. The data reduction stage aims to selectively reduce the search space while keeping all relevant data regions, whereas the query selection stage utilizes an active learning-based predictive model to iteratively improve the accuracy of the constructed exploratory query through interactions with the user. As shown, in Section IV, REQUEST is again design for structured data, therefore, it does not perform well with high dimensional unstructured data.

Example-based Search

During the past decade, a lot of interest has been generated in discovering new search interfaces beyond the traditional keyword and faceted search. For instance, in [45], a system is created to provide users with the capability to query relevant images by providing real-world photographs of the item. Based on the provided example, the system retrieves results that are visually similar to the user-provided photograph. Social media service providers such as Pinterest has designed systems that allows users to find items visually similar to the current viewing item (e.g., [46]). Recently, many studies [47], [48], [49], [50], [51] explored the problem of effectively learning the relative similarities of images. In particular, each image is mapped to a numerical vector, so the visual similarities can be captured by measuring the distance between two vectors. Leveraging these works, many enterprise visual search platforms have been developed, such as Google Similar Images, Amazon Flow, Microsoft (Bing) [52], Pinterest [46], eBay [53], and Alibaba [54]. However, all these systems still require the user to describe their desired items explicitly in some form (i.e., provide photographs that describe the desired items), before any meaningful results can be retrieved—just like traditional keywords, or query-based search systems. Therefore, these systems would fail when the user does not have an actual visual representation of the desired items, but instead a mental awareness of it.

In real-world, users often find themselves in a situation where they do not know how to properly describe their desired data or items, which is essentially the challenge that ExNav aims to address. Most importantly, ExNav is capable of working with any unstructured data types (e.g., visual, textual, and audio) as long as an embedding representation for each data element can be created. Additionally, ExNav provides the capability to dynamically adapt the exploration based on the interactive user feedback, whereas the above-mentioned systems do not allow the user to fine-tune their search session with additional feedback.

VI. Conclusion

Motivated by the challenge of reducing human effect in exploring large unstructured datasets, in this work, we proposed ExNav, a novel generic Explore-by-Examples data exploration framework for effective interactive exploration of unstructured data. ExNav effectively navigates users through the large exploration space to find relevant data items that are often undiscoverable by traditional exploration or search methods.

Our ExNav enables exploration of any unstructured data as long as an embedding representation can be obtained. In addition, we described in detail two key components of ExNav, namely, Multi-instance Space Pruning and Query Strategy, along with a set of optimization techniques that helps to improve the effectiveness of ExNav.

We implemented a prototype of ExNav and experimentally verified its performance with three real-world datasets. The results have shown that our proposed ExNav exhibits significantly better performance when compared to the state-of-the-art while achieving desired interactive performance. Specifically, ExNav can reduce users' effort by up to 92.3% (i.e., 13x less than the state-of-the-art) while still achieving the same accuracy as the state-of-the-art alternative.

Acknowledgment: We would like to thank Brian T. Nixon for his thoughtful comments on this paper. This work was partially supported by NIH award U01HL137159 and reflects only the authors' opinions.

References

[1] C. H. Teo, H. Nassif, D. N. Hill, S. Srinivasan, M. Goodman, V. Mohan, and S. V. N. Vishwanathan, "Adaptive, personalized diversity for visual discovery," *CoRR*, vol. abs/1810.01477, 2018.

[2] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, "An optimization framework for query recommendation," in *ACM WSDM*, 2010.

[3] H. Feild and J. Allan, "Task-aware query recommendation," in *ACM SIGIR*, 2013.

[4] S. Islam, C. Liu, and R. Zhou, "A framework for query refinement with user feedback," in *J. Syst. Softw., 86(6):15801595*, 2013.

[5] B. Qarabaqi and M. Riedewald, "User-driven refinement of imprecise queries," in *IEEE ICDE*, 2014.

[6] K. Dimitriadou, O. Papaemmanouil, and Y. Diao, "Explore-by-example: an automatic query steering framework for interactive data exploration." in *ACM SIGMOD*, 2014.

[7] Y. Diao, K. Dimitriadou, Z. Li, W. Liu, O. Papaemmanouil, K. Peng, and L. Peng, "Aide: an automatic user navigation system for interactive data exploration." in *PVLDB*, 2015.

[8] H. Li, C.-Y. Chan, and D. Maier, "Query from examples: An iterative, data-driven approach to query construction," in *PVLDB*, 2015.

[9] K. Dimitriadou, O. Papaemmanouil, and Y. Diao, "Aide: An active learning-based approach for interactive data exploration," *IEEE TKDE*, vol. 28, pp. 2842–2856, Nov 2016.

[10] X. Ge, Y. Xue, Z. Luo, M. A. Sharaf, and P. K. Chrysanthis, "Request: A scalable framework for interactive construction of exploratory queries," in *IEEE BigData*, 2016.

[11] E. Huang, L. Peng, L. D. Palma, A. Abdelkafi, A. Liu, and Y. Diao, "Optimization for active learning-based interactive database exploration," *PVLDB*, 2018.

[12] N. Stewart, G. Brown, and N. Chater, "Absolute identification by relative judgment." *Psychological review*, vol. 112, pp. 881–911, 11 2005.

[13] "CBC coronavirus news dataset," 2020. [Online]. https://www.kaggle.com/ryanxjhan/cbc-news-coronavirus-articles-march-26

[14] "Caltech-256 object category dataset," 2020. [Online]. Available: https://authors.library.caltech.edu/7694/

[15] "Mashup PPI dataset," 2020. [Online]. Available: http://cb.csail.mit.edu/cb/mashup/

[16] B. Settles, "Active learning literature survey," University of Wisconsin-Madison, Tech. Rep., 2009.

[17] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *ACM SIGIR*, 1994.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.

[19] Y. Yang, D. Cer, A. Ahmad, M. Guo, J. Law, N. Constant, G. H. Ábrego, S. Yuan, C. Tar, Y. Sung, B. Strope, and R. Kurzweil, "Multilingual universal sentence encoder for semantic retrieval," in *ACL*, 2020.

[20] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *ACL EMNLP*, 2014.

[21] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: large-scale information network embedding," in *ACM WWW*, 2015.

[22] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *ACM SIGKDD*, 2016.

[23] W. Hong, J. Yuan, and S. D. Bhattacharjee, "Fried binary embedding for high-dimensional visual features," in *IEEE CVPR*, 2017.

[24] B. Settles, M. Craven, and S. Ray, "Multipleinstance active learning," in *NIPS*, 2008.

[25] M. Carbonneau, E. Granger, and G. Gagnon, "Bag-level aggregation for multiple-instance active learning in instance classification problems," *IEEE TNNLS*, vol. 30, no. 5, pp. 1441–1451, 2019.

[26] W. Ren, K. Huang, D. Tao, and T. Tan, "Weakly supervised large scale object localization with multiple instance learning and bag splitting," *IEEE TPAMI*, vol. 38, no. 2, pp. 405–416, 2016.

[27] J. Zhu, J. Wu, Y. Xu, E. I. Chang, and Z. Tu, "Unsupervised object class discovery via saliency-guided multiple class learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 4, pp. 862–875, 2015.

[28] G. Quellec, G. Cazuguel, B. Cochener, and M. Lamard, "Multiple-instance learning for medical image and video analysis," *IEEE Reviews in Biomedical Engineering*, vol. 10, pp. 213–234, 2017.

[29] S. Ray and M. Craven, "Supervised versus multiple instance learning: an empirical comparison," in *ACM ICML*, 2005.

[30] D. Zhang, J. He, and R. D. Lawrence, "MI2LS: multi-instance learning from multiple informationsources," in *ACM SIGKDD*, 2013.

[31] F. Briggs, X. Z. Fern, and R. Raich, "Rank-loss support instance machines for MIML instance annotation," in *ACM SIGKDD*, 2012.

[32] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *NIPS*, 2017.

[33] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[34] R. J. Brachman, W. W. Cohen, and T. Dietterich, *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2012.

[35] Y. Xue and M. Hauskrecht, "Robust learning of classification models from noisy soft-label information." in *ECCV*, 2016.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE CVPR*, 2016.

[37] H. Cho, B. Berger, and J. Peng, "Compact integration of multi-network topology for functional analysis of genes," *Cell systems*, vol. 3, no. 6, pp. 540–548, 2016.

[38] D. Mottin, M. Lissandrini, Y. Velegrakis, and T. Palpanas, "New trends on exploratory methods for data analytics," in *PVLDB*, 2017.

[39] S. B. Roy, H. Wang, G. Das, U. Nambiar, and M. Mohania, "Minimum-effort driven dynamic faceted search in structured databases," in *ACM CIKM*, 2008.

[40] S. B. Roy, H. Wang, U. Nambiar, G. Das, and M. Mohania, "Dynacet: Building dynamic faceted search systems over databases," in *IEEE ICDE*, 2009.

[41] K. T. A. N. Niranjan Kamat, Prasanth Jayachandran, "Distributed and interactive cube exploration," in *IEEE ECDE*, 2014.

[42] D. Dash, J. Rao, N. Megiddo, A. Ailamaki, and G. Lohman, "Dynamic faceted search for discovery-driven analysis," in *ACM CIKM*, 2008.

[43] A. Kalinin, U. Cetintemel, and S. Zdonik, "Interactive data exploration using semantic windows," in *ACM SIGMOD*, 2014.

[44] E. Huang, L. D. Palma, L. Cetinsoy, Y. Diao, and A. Liu, "AIDEme: An active learning based system for interactive exploration of large datasets," NeurIPS, 2019.

[45] M. H. Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg, "Where to buy it: Matching street clothing photos in online shops," in *IEEE ICCV*, 2015.

[46] Y. Jing, D. Liu, D. Kislyuk, A. Zhai, J. Xu, J. Donahue, and S. Tavel, "Visual search at pinterest," in *ACM SIGKDD*, 2015.

[47] A. Babenko, A. Slesarev, A. Chigorin, and V. S. Lempitsky, "Neural codes for image retrieval," in *ECCV*, 2014.

[48] H. Lai, Y. Pan, Y. Liu, and S. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *ACM CVPR*, 2015.

[49] W.-J. Li, S. Wang, and W.-C. Kang, "Feature learning based deep supervised hashing with pairwise labels," in *IJCAI*, 2016.

[50] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," in *AAAI*, 2014.

[51] H. Zhu, M. Long, J. Wang, and Y. Cao, "Deep hashing network for efficient similarity retrieval," in *AAAI*, 2016.

[52] H. Hu, Y. Wang, L. Yang, P. Komlev, and L. Huang, "Webscale responsive visual search at bing," in *ACM SIGKDD*, 2018.

[53] F. Yang, A. Kale, Y. Bubnov, L. Stein, Q. Wang, H. Kiapour, and R. Piramuthu, "Visual search at ebay," in *ACM SIGKDD*, 2017.

[54] Y. Zhang, P. Pan, Y. Zheng, K. Zhao, Y. Zhang, X. Ren, and R. Jin, "Visual search at alibaba," in *ACM SIGKDD*, 2018.