Relevance- and Interface-driven Clustering for Visual Information Retrieval

Mohamed Reda Bouadjenek[a,*], Scott Sanner[b], Yihao Du[b]

[a]*School of Information Technology, Deakin University, Waurn Ponds Campus, Geelong, VIC 3216, Australia*
[b]*Department of Mechanical and Industrial Engineering, The University of Toronto, ON, Canada*

Abstract

Search results of spatio-temporal data are often displayed on a map, but when the number of matching search results is large, it can be time-consuming to individually examine all results, even when using methods such as filtered search to narrow the content focus. This suggests the need to aggregate results via a clustering method. However, standard unsupervised clustering algorithms like K-means (i) ignore relevance scores that can help with the extraction of highly relevant clusters, and (ii) do not necessarily optimize search results for purposes of visual presentation. In this article, we address both deficiencies by framing the clustering problem for search-driven user interfaces in a novel optimization framework that (i) aims to maximize the relevance of aggregated content according to cluster-based extensions of standard information retrieval metrics and (ii) defines clusters via constraints that naturally reflect interface-driven desiderata of spatial, temporal, and keyword coherence that do not require complex ad-hoc distance metric specifications as in K-means. After comparatively benchmarking algorithmic variants of our proposed approach – RadiCAL – in offline experiments, we undertake a user study with 24 subjects to evaluate whether RadiCAL improves human performance on visual search tasks in comparison to K-means clustering and a filtered search baseline. Our results show that (a) our binary partitioning search (BPS) variant of RadiCAL is fast, near-optimal, and extracts higher-relevance clusters than K-means, and (b) clusters optimized via RadiCAL result in faster search task completion with higher accuracy while requiring a minimum workload leading to high effectiveness, efficiency, and user satisfaction among alternatives.

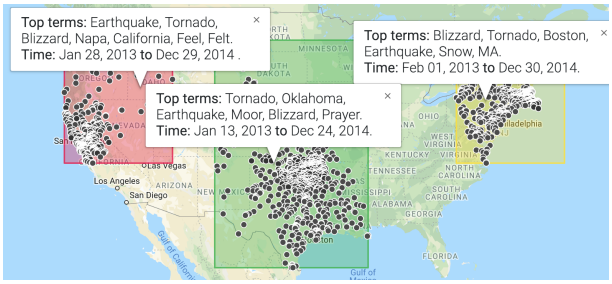
Keywords: Visual Information Retrieval; Relevance-driven Clustering; Visual Search User Study; Clustering via Filter Optimization.

1. Introduction

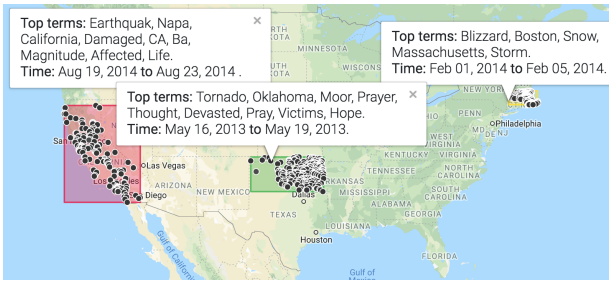
Search results of spatio-temporal data are often displayed on a map or other visual interface [1, 2, 3, 4, 5, 6]. However, given the massive volume of available information in many applications (e.g., thousands of geolocated tweets matching a query), displaying all relevant results would often result in a saturated and unreadable display [7, 8, 9].

*This work has been primarily completed while the author was at the University of Toronto.
Email addresses: reda.bouadjenek@deakin.edu.au (Mohamed Reda Bouadjenek), ssanner@mie.utoronto.ca (Scott Sanner), duyihao@mie.utoronto.ca (Yihao Du)

(a) Baseline display showing all results.



(b) K-means based clustered results.



(c) Relevance-driven clustered results.

Figure 1: (a) An interface for visual information retrieval in a multiyear Twitter corpus showing *all* geolocated tweets that match a query related to natural disasters. (b) A K-means clustered version of the *same* matching search results, but only showing the *top three* clusters of tweets delineated by bounding boxes. Since K-means uses a compound distance metric that must trade off distance in time, space, and keyword content, its clusters often include unrelated content that is further exacerbated by the fact that K-means does not take into account relevance scores of content w.r.t. the query. For these reasons, while K-means does manage to find reasonable clusters in the data, one can see that the spatial and content coherency of the clusters can be improved. (c) A relevance and interface-driven clustered version of the *same* search results given by the proposed approach in this article (again only showing the *top three* clusters, which are notably more concise than K-means). In this case, one can readily identify three well-defined natural disasters from the clusters: (i) a blizzard in Boston in February 2014, (ii) a tornado in Oklahoma in May 2013, and (iii) an earthquake in California in August 2014.

In many settings, it is natural to assume that search results cluster into spatially, temporally, and topically related content that can be aggregated and presented as a single unit rather than individual results [10]. Such approaches leverage the *cluster hypothesis* of Information Retrieval (IR) [11, 12, 13, 10], which posits that documents in the same cluster should behave *similarly* with respect to information needs.

For an example visual search use case, consider the task of searching a multiyear Twitter corpus for content related to natural disasters.[1]. Visual Twitter search is chosen here due to the availability of high volume spatio-temporal data and it's general familiarity to our test subjects. A conventional IR list ranking approach based only on the textual content of the tweets would flood the user with an extremely long list of tweets — such simple ranked listings fail to effectively impart the spatial-temporal distribution of search

[1]While this example involves visual Twitter search, the methods defined in this article are not specific to this application but are generally intended for any user task involving a query-driven visual search interface with a large volume of matching search results and content that naturally clusters along spatial, temporal, and keyword content dimensions

results. However, as shown in Figure 1a, a typical spatial-temporal-content visualization approach that would provide all matching tweets in a map-based display will similarly take the user a large amount of time to sift through; while content is now spatially distributed, there is still too much matching content to sift through. To help reduce this visual information overload, filtering and faceted search [14, 15, 16, 17] can be used here to help the user manually narrow the large set of tweets using filter settings defined for each tweet aspect (location, posting time, keywords). Although such navigation systems are commonly used and hence comprise a baseline in our user study, a large amount of effort is still required on behalf of the user to manually read through results and adjust filters appropriately.

To deal with this latter problem and ease the task of browsing search results, a clustered results display like that shown in Figure 1b can be used to restrict the displayed information such that similar tweets appear together. Most existing work on aggregation for visual search that has sought to exploit the cluster hypothesis has focused on K-means and related unsupervised clustering methods [18, 8, 2, 1, 19, 20, 21, 22] that do not necessarily guarantee that clusters of matching search results are highly relevant. Moreover, the use of clustering algorithms such as K-means requires the design of a complex distance metric; for example, consider that space is often measured by Euclidean distance while keyword content is often measured by cosine distance and both of these distances need to be combined into a *single* distance metric for K-means. Such ad-hoc metric specifications do not necessarily guarantee the coherence of clusters from a visual, temporal and keyword content perspective. In contrast, as demonstrated in Figure 1c, we will contribute a clustering algorithm that is actively aware of the relevance probability of each tweet to the search query and thus can automatically generate highly relevant clusters covering a large fraction of *relevant* content while explicitly optimizing for interface-driven desiderata of spatial, temporal, and keyword coherence *without* requiring any ad-hoc specification of a complex distance metric that requires trading off distances in each dimension.

In this article, we realize the vision discussed above and demonstrated in the example of Figure 1c by addressing clustering for visual search in a novel relevance- and interface-driven optimization framework. Specifically, we make key novel contributions that span both the user-focused design of a novel visual-search driven interface as well as numerous technical innovations required to realize this design. We also contribute quantitative and qualitative evaluations of both our technical contributions and their overall benefit to end users in a visual search task for Twitter.

We summarize these numerous contributions as well as an outline of the article as follows:

1. To better satisfy end-user task needs for clustering in visual search interfaces, we present a novel *relevance-driven clustering objective* that extends standard information retrieval metrics to clustering. Specifically, in light of relevance uncertainty, we derive *expected metrics for precision and recall of clusters*, but ultimately argue that a good cluster must balance both and thus focus on a derivation of *expected F1-score (EF1)* of cluster relevance as our key objective. Two key features of EF1 are that (a) it automatically extracts coherent clusters in terms of space, time, and content for presentation in a visual search interface and that (b) optimizing it does not require the specification of complex ad-hoc

distance metrics required by other unsupervised clustering algorithms such as K-means.

2. Through a series of transformations, we demonstrate that the globally optimal solution to EF1 maximization of clusters can be cast as a Mixed Integer Linear Program (MILP), which is unfortunately NP-hard and thus computationally expensive to solve. To improve the algorithmic efficiency of optimization, we present two algorithms: Greedy and Binary Partitioning Search (BPS). Referring to our Relevance-driven Clustering Algorithm as RadiCAL, this leads to three variants: RadiCAL-MILP, RadiCAL-Greedy, and RadiCAL-BPS. We quantitatively evaluate and compare all RadiCAL variants and K-means on a search-driven tweet clustering task and demonstrate that RadiCAL-BPS provides the best overall tradeoffs in terms of performance and efficiency.
3. Returning to our end-user visual search task motivation, we conclude the experimental evaluation of this work with a user study to evaluate whether this new relevance-driven clustering method improves human performance in comparison to K-means clustering and a multiple filter search baseline.[2] Our results show that clusters derived in our relevance- and interface-driven optimization framework result in faster search task completion with higher accuracy while requiring a minimum workload leading to high effectiveness, efficiency, and user satisfaction among alternatives. These results coincide with our offline evaluation that also demonstrate the superiority of our relevance-driven clustering approach over competing methods.

As a final remark, all the algorithms described throughout this paper have been integrated into a tool called Visual Twitter Information Retrieval (Viz-TIR) [23]. Although RadiCAL-BPS presented here is briefly described in [23], in this article, we provide its detailed description along with its simpler RadiCAL-Greedy variant and the derivation of the *optimal* RadiCAL-MILP solution; we also comparatively benchmark all RadiCAL variants and K-Means. We further provide extensive offline and human user evaluation results and analysis that substantially expands on the results presented in [23].

2. Related work

There is a substantial body of research related to visual search and clustering. Below, we review the major works related to clustering, filtering, optimization, and visualization in information retrieval. Because our research focus is not specifically on visual Twitter search or disaster informatics (this was simply a use case amenable to user experimentation as discussed previously), these topics are too narrowly focused to

[2]While there are a large number of unsupervised clustering algorithms in the literature, we had limited user interaction time in our user study and thus could only choose one clustering algorithm for comparison in addition to the non-aggregation baseline. We chose K-means since it is arguably the most commonly used clustering algorithm – not only in general, but also specifically in our coverage of related work on clustering in information retrieval and visual search.

warrant an exhaustive discussion here, though many related citations occur in the topics discussed below.

Spatio-temporal clustering: A lot of research has been done on the clustering of spatio-temporal data points [24, 25], and this has been done for different applications including crime discovery [26], Twitter data mining [27, 28, 29, 30], geo-tagged photo exploration [31, 32], traffic accident monitoring [31], and epidemiology monitoring [33]. Most proposed techniques are based on clustering methods such as K-means [34], BIRCH [35], OPTICS [36], or DBSCAN [37]. DBSCAN is a widely used method for finding arbitrarily shaped clusters of spatial points based on the density of points, which has been extended to temporal data in ST-DBSCAN [38]. More recently, Choi and Chung [39] proposed a modified version of the K–means clustering algorithm for spatio-textual as opposed to spatio-temporal data.

We note that all of these clustering methods fail to jointly address our goals for visual search clustering as stated in the introduction. Namely, these methods (i) ignore relevance signals (beyond the initial search), (ii) ignore the *joint combination* of spatial, temporal, and keyword constraints, and/or (iii) ignore definitions of clusters that pertain to their presentation in a visual display medium, all of which are key jointly intertwined contributions of the clustering approach proposed in this work.

Clustering in IR: Clustering is an active research field as evidenced by recent work [39, 40, 41, 42, 43, 44] and even the specialization of clustering for IR remains an active area of research [45, 46]. Clustering in IR has been used in a variety of applications, which differ in terms of the set of elements clustered and the overall aim of clustering. Clustering of search results themselves has been investigated for more effective information presentation to users [46, 47, 48, 49]. For example, Kurland *et al* [50, 51] use clustering of top search results to improve relevance scoring models for ranking. On the other hand, collection clustering has been used for effective information presentation and for exploratory browsing [45, 52, 53], for improving search results [54, 55], and for speeding up search [11, 56, 57]. In yet another vein, Scatter-Gather, which consists of repetitively clustering and selecting clusters, has been proposed as an alternative user interface to explore elements without using explicit queries [58, 59].

When considering our search-based clustering needs in this article, all of these methods (i) do not explicitly use the relevance signal during cluster optimization to ensure high relevance of extracted clusters, and (ii) do not specifically formulate clusters in terms of spatial, temporal, and content constraints to ensure coherence and succinct presentation in a visual search display. Both of these requirements are addressed in our proposed contributions.

Filtering and Faceted Search in IR: Belkin and Croft [60] defined information filtering as a counterpart to IR, albeit with a few key differences. Namely, information filtering often occurs in the context of a long-term standing interest (represented implicitly through a relevance measure), as well as continuing interaction with the filtering system over a long period of time. Most work on information filtering displays has so far focused on *unsupervised* approaches such as dynamic adjustment of parameters [18, 19, 61], (hierarchical) clustering [62, 1, 22], topic classification [2, 63, 64], and layout algorithms [21, 65, 66, 67].

Since our cluster definition is based on constraints, it may be natural to think of our clusters as Belkin and Croft's information filters for interactive visual search. However, the similarity more or less ends there. In this work, we have an explicit query that drives construction of our filters. Further, we directly optimize our filter settings w.r.t. a relevance-based objective to maximize the expected F1-Score given a probabilistic measure of relevance. While these techniques may be used to extend work in information filtering, no existing information filtering work performs the same relevance-driven cluster (or filter) optimization that we propose in this work.

Separately from Belkin and Croft [60], others have defined *filtering and faceted search* methods [14, 15, 16, 17] — the idea that one should be able to restrict the content shown by adjusting multiple filters to restrictions on different dimensions of meta-data (e.g., time stamp or location). While our methods arguably build on ideas in multiple filter search and we compare to a filtered search baseline, the key distinction is that existing filtered search has focused on the user interface design and user studies, whereas our work focuses explicitly on automatically extracting clusters (where an individual cluster corresponds to a setting of multiple filters) to maximize relevance-driven optimality criteria.

Explicit Optimization of IR Metrics: In this work, we focused on clustering as an explicit optimization of an IR-derived metric. While no other existing work has proposed an expected F1-Score relevance-driven optimization approach to clustering as we do here, it is still worthwhile to explore what other explicit optimization approaches have been taken in IR. In that context, Wang and Zhu [68] proposed to use an expected score approximation to optimize Average Precision, Discounted Cumulative Gain, and Reciprocal Rank. However, the authors did not propose a way to optimize Boolean metrics such as recall and F1-Score, which are critical for our cluster optimization objective. Separately, machine learning has been explored to optimize different metrics such as NDCG or MAP through Learning to Rank (L2R) [69]. However, L2R cannot be applied in our cluster optimization problem because we do not have labeled data to train with — while we have a relevance signal, true cluster labels are not known for any data. Moreover, the task we address is to find cluster settings that optimize an expected Boolean metric of expected F1-Score – not to optimize metrics for ranking.

3. Framework and notation

In this section, we first define the problem we address and then the mathematical notation we use.

3.1. Problem definition

To define the "ingredients" we have for the visual search clustering problem, we begin by adopting the standard information retrieval (IR) setting for both querying and retrieving information elements (e.g., documents or tweets) [69], but in a visual search interface capable of displaying clusters as shown in Figures 1b and 1c. Specifically, assuming a given corpus of text-based information elements with both time stamp and 2D spatial location meta-data, the retrieval process can be initiated as follows: The user first specifies a

Boolean "or" query consisting of a list of keywords. For each retrieved document, a score is then computed indicating the probability of relevance to the user's information need as specified via their query.[3]

Our next task is to visually cluster the search results according to the following criteria. As motivated in the introduction, a key desiderata is that we do *not* want our clustering solution to require the specification of a *joint* distance metric between two information elements in terms of text content, time, and space. Instead, we only assume the search results have three display attributes that can be used to define *coherent* clusters in terms of spatial, temporal, and topical criteria. We define a *coherent cluster* as a group of elements that are topically similar to each other (i.e., similar text content) and that are similar in both their time and space dimensions. We remark that coherency in the spatial dimension forms a key requirement for clusters that can be easily (i.e., compactly) visualized. Specifically, we consider the following attribute constraints to define a cluster:

- **Space:** limits clusters content with 2D spatial annotations (e.g., latitude and longitude) according to four parameters for the upper left and lower right bounding box coordinates. These cluster constraints define the bounding box that is visually displayed to the user, cf. Figure 1c.
- **Time:** limits cluster content with time-stamps according to two parameters for the lower and upper time bound. Time can be displayed via cluster labels, and/or through settings of a time slider in the interface.
- **Keyword (or Discrete Attribute):** limits cluster content with text annotations according to included or excluded keywords (or general discrete attributes of an information element). Explicitly included and excluded keywords can be used to label the cluster.

We note that this clustering work is not limited to these three display attributes – any continuous or discrete cluster attributes that naturally constrain the search results can be accommodated by our framework. Nonetheless, we believe time, space, and content constitute three of the most common information display attributes in practice and hence are the ones we focus on in this work.

Given the above ingredients, cluster specification constraints, and desiderata for our search-based visual clustering problem, we now have three research questions to answer in this article: (1) How can we formulate an optimization objective to extract clusters that satisfy all criteria above? (2) How can we efficiently optimize this objective for use in real-time visual search on large corpora? (3) How we can we evaluate the effectiveness of these visual search clustering algorithms compared to commonly used alternatives through both offline evaluations and user studies?

[3]In this paper, we adopt an IR interpretation of *relevance*, which refers to how well a document or a cluster meets the user's *information need*. An information need is defined as the information that satisfies a conscious or unconscious need of the user and is formally expressed by the user's keyword-based query [69]. In this paper we will specifically use a *language model* definition of probabilistic relevance w.r.t. a user's Boolean "or" query [70].

3.2. Mathematical Notation

With the cluster definitions above, we now define formal mathematical notation used throughout the remainder of the article:

- An information element j (i.e., a search result) may have three types of associated metadata: (i) position coordinates (x_j, y_j), (ii) a timestamp t_j, which may represent the creation date of j, and (iii) textual content, which is composed of a set of unique terms $\{t_1, \ldots, t_n\}$ of size n (to reduce notational clutter, we assume the element j containing these terms will be clear from context).

- Three variables $I_q(j) \in \{0, 1\}$, $B_q(j) \in \{0, 1\}$ and $S_q(j) \in [0, 1]$ are associated with each information element j and a query q: $I_q(j)$ is an indicator referring to whether an element j is retrieved and displayed (true$=1$) given a query q; $B_q(j)$ is a Boolean random variable indicating the (ground truth) relevance of an element j (relevant $= 1$) w.r.t. a query q; $S_q(j)$ is a relevance score indicating the *probability* relevance of an element j w.r.t. a query q. $B_q(j)$ follows a *Bernoulli* distribution with parameter $S_q(j)$, and hence, the expectation of $B_q(j)$ *is* $S_q(j)$, i.e., $\mathbb{E}[B_q(j)] = S_q(j)$.

- We label GC as the global set of all information elements j with total size $|GC| = m$. E_q is the set of retrieved information elements that match a user query q, where $E_q \subseteq GC$. We use E_q^* to refer to further subsets of elements of clusters, i.e., $E_q^* \subseteq E_q$. Note that $|E_q|$ is the count of retrieved $I_q(j)$ among the global collection GC. Therefore, we have $|E_q| = \sum_{j=1}^{m} I_q(j)$.

- We label the set of ground truth relevant information elements for a query q as the relevant set RS_q consisting of $|RS_q|$ elements. Note that $|RS_q|$ is the count of relevant $B_q(j)$ among the global collection GC. Therefore, we have $|RS_q| = \sum_{j=1}^{m} B_q(j)$.

4. Relevance-driven clustering

First, we proceed to motivate and derive a new *expected* F1-Score we use to optimize for cluster extraction. Following this, we describe two efficient greedy algorithms to (approximately) optimize it.

4.1. Motivating and Deriving Expected F1-Score (EF1)

Existing multidimensional clustering methods such as K-means commonly used in search visualization largely ignore relevance signals. In contrast, our objective in this article is to take an "information retrieval first" approach, i.e., to reconceive information retrieval if the goal was to present results as visual clusters as opposed to the more usual ranked list. Because clusters are the manner by which we return search results and clusters correspond to a Boolean selection of information elements, we will argue in this section that (expected) F1-Score of clusters is the only standard Boolean relevance criteria that balances all of our cluster desiderata and is hence the information retrieval objective we should optimize.

To proceed with the formal derivation, we adopt the Boolean relevance framework in information retrieval [69] and thus assume that an information element j has a ground truth relevance assessment $B_q(j)$ w.r.t. q available at *evaluation time*. Because clustering implies a Boolean retrieval model (clusters either contain or do not contain elements) and we have a probabilistic estimate of relevance $S_q(j)$, we propose to evaluate *expected* variants of standard precision, recall, and F1-score of these clusters.[4,5]

However, as standard for both precision and recall, we note that precision and recall alone can be trivially optimized by undesired solutions. That is, the cluster that selects all information elements (i.e., all time, all space, no excluded keywords) would trivially maximize (expected) recall. Similarly, the cluster that selects the highest probability singleton information element would maximize expected precision. *This leaves expected F1-score as the only of these three objectives commonly used in Boolean information retrieval that does not have an undesired solution.*

To formally define expected F1-Score, we first begin with definitions of expected precision and recall. Recalling our previous definitions, given a set of information elements E_q that match a user query q and a relevant set RS_q, the precision of E_q is defined as follows:

$$P(E_q) = \frac{\sum_{j \in E_q} B_q(j)}{|E_q|} = \frac{\sum_{j=1}^{m} B_q(j) I_q(j)}{\sum_{j=1}^{m} I_q(j)} \tag{1}$$

Given that $B_q(j)$ is a Boolean random variable, we can take the expectation of $P(E_q)$ leading to the following definition of *expected precision* $EP(E_q)$:

$$EP(E_q) = \mathbb{E}_{\mathbb{S}}\left[\frac{\sum_{j=1}^{m} B_q(j) I_q(j)}{\sum_{j=1}^{m} I_q(j)}\right] = \frac{\sum_{j=1}^{m} \mathbb{E}_{\mathbb{S}}[B_q(j)] I_q(j)}{\sum_{j=1}^{m} I_q(j)} = \frac{\sum_{j=1}^{m} S_q(j) I_q(j)}{\sum_{j=1}^{m} I_q(j)} \tag{2}$$

Similarly the recall of a retrieved set $R(E_q)$ is defined as:

$$R(E_q) = \frac{\sum_{j \in RS_q} B_q(j)}{|RS_q|} = \frac{\sum_{j=1}^{m} B_q(j) I_q(j)}{\sum_{j=1}^{m} B_q(j)} \tag{3}$$

Taking a 1st order Taylor expansion, we have the following expectation approximation $\mathbb{E}(X/Y) \approx \mathbb{E}(X)/\mathbb{E}(Y)$ for two dependent random variables X and Y [72]. Hence, we can now define an *approximated expected recall* as follows:

$$ER(E_q) = \mathbb{E}_{\mathbb{S}}\left[\frac{\sum_{j=1}^{m} B_q(j) I_q(j)}{|RS_q|}\right] \approx \frac{\sum_{j=1}^{m} \mathbb{E}_{\mathbb{S}}[B_q(j)] I_q(j)}{\sum_{j=1}^{m} \mathbb{E}_{\mathbb{S}}[B_q(j)]} = \frac{\sum_{j=1}^{m} S_q(j) I_q(j)}{\sum_{j=1}^{m} S_q(j)} \tag{4}$$

[4]We remark that since all existing clustering algorithms for information retrieval reviewed in Section 2 are unsupervised and thus focus on minimizing co-similarity of all retrieved documents within a cluster, we believe that our alternative Boolean relevance-based definitions of optimal clusters are a novel and distinct contribution to clustering in information retrieval.

[5]We note that we are not the first to consider probabilistic or expected variants of Boolean metrics. One notable work by Goutte and Gaussier [71] proposes a probabilistic re-interpretation of Boolean metrics assuming the availability of ground truth relevance since their primary purpose is to compute a Bayesian posterior estimate of Precision and Recall at evaluation time. However, in our case, we are not using the expectation for experimental evaluation purposes but rather for computing Boolean metrics under probabilistic uncertainty over the relevance of each document that occurs in the *absence of ground truth* at search retrieval time. Hence, we have a very different use and definition for our variants of these expected Boolean metrics.

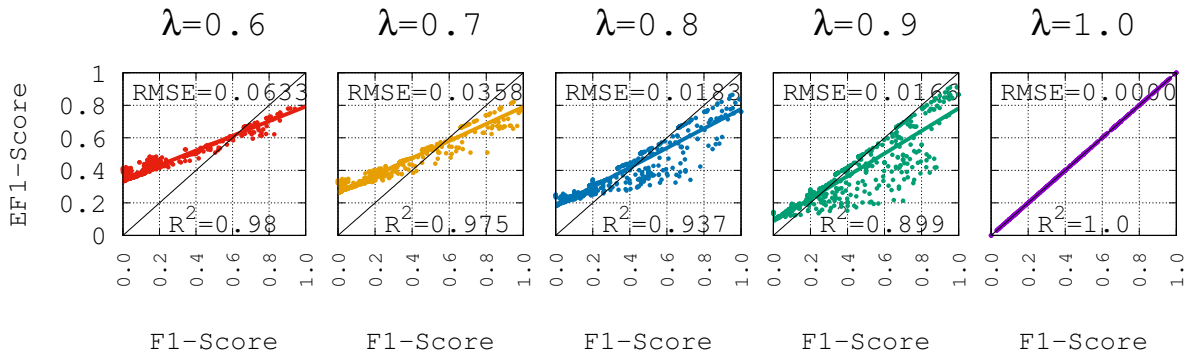


Figure 2: Scatterplot with best fit linear regression showing predicted EF1-Score vs. ground truth F1-Score as the amount of noise λ varies from 0.6 (high noise) to 1.0 (no noise). EF1 correlates with F1, hence ranking documents similarly to F1.

Finally, we define the *approximated expected F1-Score* (EF1) using the *expected precision* and the *approximated expected recall* as follows:

$$EF1(E_q) \approx \frac{2 \times EP \times ER}{EP + ER} = \frac{2 \times \sum_{j=1}^{m} S_q(j) I_q(j)}{\sum_{j=1}^{m} I_q(j) + \sum_{j=1}^{m} S_q(j)} \tag{5}$$

We focus on F1-score in this paper, however expected F_β scores that vary the relative weight of the precision and recall components according to β follow directly from the above definitions.

To validate that EF1 provides a strong surrogate metric for F1 in the absence of ground truth relevance judgments, we show a scatterplot of ground truth F1 scores vs. EF1 scores for sets of 100 elements in Figure 2. Specifically, for each element set, we set the $\{0, 1\}$ relevance indicator $B_q(j)$ for element j uniformly randomly and then assign the relevance probability $S_q(j)$ according to a noisy corruption of the ground truth: $S_q(j) = \lambda B_q(j) + (1 - \lambda)\text{rand}()$, where rand() is a random noise value chosen with uniform distribution in the range $[0, 1]$, and λ is a weighting parameter ($0.5 \leq \lambda \leq 1$) that controls the signal-to-noise ratio in the final probability value; $\lambda = 1.0$ represents no noise while $\lambda = 0.6$ represents high noise. Here we can see that maximizing EF1 score is strongly correlated with maximizing F1 score across a wide range of noise levels. Specifically, while the EF1 and F1 scores are not perfectly calibrated along the diagonal, there is a clear linear correlation indicating that EF1 and F1 will both rank information elements in a similar order. We will further study the effect of a noisy classifier in Section 6.

Finally, we address the key question of whether the EF1 objective should include additional coherence criteria. In short, we argue that coherence corresponds to "tight" constraint settings in all dimensions (spatial, temporal, keyword); while Precision and Recall arguably lead to incoherent clusters (respectively, too small or too large), F1-Score balances these two to return moderately sized clusters. If an F1-Score cluster shrinks unnecessarily, it's Recall component would decrease and make it suboptimal, while if it expands unnecessarily, it's Precision component will decrease and also make it suboptimal. Hence, optimizing clusters for EF1 does correspond to some *locally optimal* notion of temporal, semantic, and spatial coherence when considering expansions or contractions of the selected cluster. These claims of coherence for relevance-driven clustering based on EF1 are directly evidenced when we observe that the relevance-driven clusters of

Figure 1c are much more "tight" in terms of time span (a few days to a month), spatial extent (well localized), and top keyword content (words are coherent) than K-means shown in Figure 1b which has non-localized spatial extent, unnecessarily large time spans of 11 months or more, and incoherent top keywords in a single cluster ("Blizzard", "Earthquake", and "Tornado" together in the right-most cluster).

4.2. Greedy relevance-driven clustering

We now specify an algorithm to *efficiently* optimize clusters for relevance according to the previously defined EF1 metric. Specifically, a single cluster E_q^* is specified as all information elements $\{j \in E_q^* | I_q(j) = 1, j \in E_q\}$ in the intersection of (i) keyword inclusion or exclusion, (ii) time interval, and (iii) spatial bounding box constraints. Given an estimated probability of relevance of each information element in the cluster $S_q(j)$, we can compute the EF1 of the cluster defined by constraints (i)–(iii) according to (5). Through a series of transformations, this EF1 optimization problem can be reduced to an *optimal* Mixed Integer Linear Program (MILP) solution, which we outline in Appendix A; however, we remark that a MILP-based approach is NP-Hard and only practical in real-time for small element sets, thus we can only use it for benchmarking other algorithms in Section 6. Hence, in pursuit of a more tractable solution, in the following we describe how to *greedily* optimize the parameters of each constraint to approximately optimize clusters according to EF1.

In greedy optimization, we would have the option to start with a singleton element cluster and expand, or start with a cluster including all elements and prune. While the former has a non-deterministic choice of which singleton to start with, the latter has an unambiguous initial starting condition. Thus we choose the latter pruning approach starting with initial spatial bounding box, time interval, and keyword constraints set to include *all* of E_q.

4.2.1. Greedy Keyword Selection algorithm

Given a set of candidate information elements matching a query, this algorithm greedily selects a set of keywords to exclude (i.e., prune) to maximize EF1.

Formally, the algorithm aims to select an optimal subset of k terms $T_k^* \subset V$ (where V is a vocabulary of keywords for the document corpus) to exclude elements containing these keywords to optimize EF1. This is achieved by building T_k^* in a greedy manner by choosing the next optimal term t_k^* given the previous set of optimal term selections $T_{k-1}^* = \{t_1^*, \ldots, t_{k-1}^*\}$ (assuming $T_0^* = \emptyset$) using the following selection criterion

$$t_k^* = \underset{t_k \notin T_{k-1}^*}{\arg\max} [EF1(E_q^* \text{ that don't contain } \{t_1^*, \ldots t_k^*\})] \tag{6}$$

that terminates at k when no T_{k+1}^* can improve the EF1 of E_q^*.

4.2.2. Greedy Time Selection algorithm

The idea behind the time-based greedy selection algorithm is to start with the maximal time range and greedily contract it to an open sub-interval of time $(t_{\text{start}}, t_{\text{end}})$ that maximizes EF1. In this case, given a list of currently selected elements $E_q^* = \{j_{t_1}, \ldots, j_{t_n}\}$ ordered by timestamp, where j_{t_i} for $1 \leq i \leq n$ is

Algorithm 1: Binary Partition Search (BPS) Algorithm

input : A set of ordered elements $E_q = \{j_{v_1} \dots j_{v_n}\}$ (ordered with respect to v_k);

output: A value v for the (possibly contracted) upper bound;

1 $v_{min} = v_1$; $v_{max} = v_n$; $v_{mid} = \frac{v_1+v_n}{2}$;

2 **while** $v_{min} \neq v_{mid} \neq v_{max}$ **do**

`/* Note: the initial index for EF1 comparison is always` v_1 `-- only the upper bound is contracted, hence the reason why BPS is called once for ascending and descending order of each dimension. */`

3 **if** $[EF1(\{j_{v_1} \dots j_{v_{mid}}\}) \geq EF1(\{j_{v_1} \dots j_{v_{max}}\})]$ **then**

4 $v_{max} = v_{mid}$; $v_{mid} = \frac{v_{min}+v_{mid}}{2}$;

5 **else**

6 $v_{min} = v_{mid}$; $v_{mid} = \frac{v_{min}+v_{max}}{2}$;

7 **end**

8 **end**

9 **return** v_{mid};

the ith information element in this order with timestamp t_i, we propose two different greedy approaches to iteratively contract (i.e., prune) the time window of E_q^* in order to maximize EF1:

(a) Naive Greedy algorithm: Let $t_{\min}$ and $t_{\max}$ respectively correspond to the current minimum and maximum timestamps in E_q^*. If setting $t_{\text{start}} = t_{\min}$ improves the EF1 of E_q^* then this lower bound contraction is accepted. Similarly if setting $t_{\text{end}} = t_{\max}$ improves the EF1 of E_q^* then this upper bound contraction is accepted. This repeats until no lower or upper bound contraction improves EF1.

(b) Binary Partition Search (BPS) algorithm: Large datasets will cause the previous Greedy algorithm to take a large number of iterations to terminate. A more efficient way to address this problem is to use the binary partitioning search (BPS) subroutine shown in Algorithm 1. Consider a list of n information elements; instead of removing a single element j at a time requiring $O(n)$ iterations, BPS leverages an approach motivated by binary search over the current elements in E^* to efficiently obtain a new bound that improves EF1 score in $O(\log_2 n)$ iterations.

From an implementation perspective of the BPS approach for time selection, the algorithm first sorts the current E_q^* according to time stamps in increasing order (line 1 of Algorithm 2). The BPS algorithm then calls the BPS subroutine (line 2 of Algorithm 2) to find a new upper bound if it improves the EF1 score through an approach motivated by binary search. The resulting E_q^* (line 3 of Algorithm 2) is then sorted according to time stamps in decreasing order (line 4 of Algorithm 2) and the same BPS strategy is used to find a new lower bound if it improves the EF1 score (lines 5-6 of Algorithm 2). While the binary search in this approach does not check all contractions and thus serves as an approximation to the Naive Greedy

Algorithm 2: BPS Time Bound Contraction Algorithm

input : A set of retrieved elements E_q^*;

output: A set of elements $E_q^{*'}$ (such that $E_q^{*'} \subseteq E_q^*$);

1 Sort E_q^* in **increasing timestamp** order;

2 t_{end}^{best} = BinaryPartitionSearchE_q^*);

3 $E_q^* \leftarrow \{j \in E_q^* | t_j < t_{\text{end}}^{best}\}$;

4 Sort E_q^* in **decreasing timestamp** order;

5 t_{start}^{best} = BinaryPartitionSearch(E_q^*);

6 $E_q^* \leftarrow \{j \in E_q^* | t_{\text{start}}^{best} < t_j < t_{\text{end}}^{best}\}$;

7 **return** E_q^*;

approach, it is substantially faster ($O(\log_2 n)$ instead of $O(n)$) and a reasonable approximation, *especially* if the EF1 score changes relatively monotonically as the time bounds are contracted from each side.

4.2.3. Greedy Spatial Selection algorithm

The aim of this algorithm is to return coordinates $[(x_{min}, y_{min}), (x_{max}, y_{max})]$ representing the EF1 maximizing spatial bounding box represented by the lower and upper bound coordinates – respectively (x_{min}, y_{min}) and (x_{max}, y_{max}). This 2D spatial interval contraction problem is similar to the previous 1D problem of finding the best time window. Therefore, the two algorithms described above (Greedy and BPS) can be adapted for this problem by first applying each algorithm on the x-axis to determine the best (x_{min}, x_{max}) (lines 1-6 of Algorithm 3), then on the y-axis to determine the best (y_{min}, y_{max}) (lines 7-12 of Algorithm 3).

4.2.4. Overall Relevance-driven Clustering (RadiCAL) algorithm

To obtain a cluster E_q^* combining the above (i) keyword, (ii) time, and (iii) spatial constraints, we propose a greedy round-robin algorithm, which at each iteration applies the selection pruning algorithms for (i), (ii), and (iii) in order. Iterations terminate when no selection algorithm can unilaterally improve EF1 and the final cluster is returned.

Finally, we note that our **R**elev**a**nce-**d**ri**v**en **C**lustering **AL**gorithm (**RadiCAL**) can use the Greedy keyword selection algorithm with either the naive Greedy time and spatial selection algorithms, which we refer to experimentally as **RadiCAL-Greedy**, or the BPS time and BPS spatial variants, which we refer to experimentally as **RadiCAL-BPS**.

4.3. Multiple Cluster Selection Wrapper

In practice, a single cluster of information elements E_q^* chosen by the previously described algorithms will provide the user with *one* temporal, spatial, and content coherent cluster covering one information perspective. However, just as K-means allows one to select the number of clusters K, let us assume that

Algorithm 3: BPS Spatial Bound Contraction Algorithm

input : A set of retrieved elements E_q^*;

output: A set of elements $E_q^{*'}$ (such that $E_q^{*'} \subseteq E_q^*$);

Sort E_q^* in **increasing order** w.r.t. the location **x-axis**;

$x_{\max}^{best} = \text{BinaryPartitionSearch}(E_q^*)$;

$E_q^* \leftarrow \{j \in E_q^* | x_j < x_{\max}^{best}\}$;

Sort E_q^* in **decreasing order** w.r.t. the location **x-axis**;

$x_{\min}^{best} = \text{BinaryPartitionSearch}(E_q^*)$;

$E^* \leftarrow \{j \in E^* | x_{\min}^{best} < x_j < x_{\max}^{best}\}$;

Sort E^* in **increasing order** w.r.t. the location **y-axis**;

$y_{\max}^{best} = \text{BinaryPartitionSearch}(E^*)$;

$E^* \leftarrow \{j \in E^* | x_{\min}^{best} < x_j < x_{\max}^{best} \wedge y_j < y_{\max}^{best}\}$;

Sort E^* in **decreasing order** w.r.t. the location **y-axis**;

$y_{\min}^{best} = \text{BinaryPartitionSearch}(E^*)$;

$E^* \leftarrow \{j \in E^* | x_{\min}^{best} < x_j < x_{\max}^{best} \wedge y_{\min}^{best} < y_j < y_{\max}^{best}\}$;

return E^*;

we wanted to show $K = 3$ clusters extracted through relevance-driven optimization methods as shown in Figure 1c. Here, we provide a greedy approach for providing a ranked list of such clusters that works with any of the previously defined algorithms – **RadiCAL-Greedy** or **RadiCAL-BPS**.

The algorithm itself is quite simple and simply wraps the algorithm of Section 4.2.4. After the first cluster is produced, all selected elements in that cluster have their scores $S_q(j)$ zeroed out. The relevance-driven clustering algorithm is then run again, where it will inherently focus on a different content set. As each cluster is added (up to a user-defined limit K), coverage of high relevance content monotonically improves.

5. Experimental setup

We now describe the dataset and baselines to be used by our experimental evaluation in Sections 6 and 7.

5.1. Dataset description

The scenario we have chosen to evaluate our algorithms is related to the detection of natural disasters discussed in a collection of tweets chosen due to the availability of high volume spatio-temporal data and it's general familiarity to our human test subjects. We started with a corpus of approximately 1 billion tweets crawled from the Twitter streaming API during 2013 and 2014 [73] with the following restrictions: (1) the dataset was restricted to users located within the US, (2) non-English tweets were filtered out, (3) we extracted tweets related to the 12 actual natural disasters described in Table 1 – which are temporally and geographically disjoint – to use as ground truth clusters, and (4) we removed tweets related to other known

Table 1: Details of the events included in the dataset.

	Type	Location	Date	#tweets
1	Flood	Colorado	Sep, 2013	100
2	Storm	Florida	June, 2013	181
3	Earthquake	California (L.A.)	Mar, 2014	98
4	Earthquake	California (Napa)	Aug, 2014	206
5	Tornado	Oklahoma	May, 2013	319
6	Hurricane	North Carolina	July, 2014	98
7	Blizzard	New York (NYC)	Feb, 2014	243
8	Blizzard	New York (Buffalo)	Nov, 2014	99
9	Blizzard	Massachusetts (Boston)	Feb, 2014	201
10	Drought	California	Dec, 2013	100
11	Tornado	Mississippi	Feb, 2013	50
12	Flood	Michigan	Aug, 2014	49

natural disasters, which was necessary to create unambiguous correct answers for purposes of our user study. The final dataset[6] consists of 1,744 positive examples (tweets related to the 12 natural disasters we selected) as well as 34,411 negative examples (other tweets). We remark that the tasks of identifying each of the 12 natural disasters in the dataset were chosen to be of comparable difficulty — or, at the very least, there was no explicit intent to curate easy vs. difficult tasks since algorithm order and assignment of natural disasters to algorithm trials are necessarily randomized in the user study.

5.2. Baselines description

Our experiments use the following two baseline clustering algorithms:

Optimal solution: To benchmark the performance of **RadiCAL-Greedy** and **RadiCAL-BPS** on small datasets, we use an exact Mixed Integer Linear Progamming (MILP) optimization-based formulation to maximize EF1. In brief, the formulation of EF1 in (5) can be transformed into a *fractional* MILP formulation with constraints corresponding to each of the cluster attribute selection criteria (space, time, and content). The parameters of these constraints are then chosen to optimize the EF1 objective. While there are no direct solvers for fractional MILPs, we can transform the problem into a pure MILP formulation using the Charnes-Cooper method [74] and Glover linearization method [75] for which we have optimal (albeit slow) solvers. A

[6] `https://github.com/D3Mlab/viz-ir/tree/master/twitter_dataset`

detailed description of this optimal solution is given in Appendix A, referred to as **RadiCAL-MILP**.

***K*-means clustering:** We use the *X*-means [76] variant of *K*-means as a baseline method to **cluster matching search results**. *X*-means is a simple extension of *K*-means [34] that automatically determines the number of clusters. Starting with only one cluster, the *X*-means wrapper applies after each run of *K*-means, making local decisions about which subset of the current centroids should split themselves in order to better fit the data. In order to provide *X*-means with spatial, temporal, and content coherence, the distance metric we have used for *X*-means is a linear combination of the following: (i) the Euclidean distance of time, (ii) the Euclidean distance of location, and (iii) the cosine distance of the textual content. This distance metric is formally defined as follows:

$$d(i,j) = \alpha \times [\text{time dist.}] + \beta \times [\text{location dist.}] + \gamma \times [\text{text cosine}] \tag{7}$$

where α, β, and γ are weights that sum to 1, and set respectively to 0.1, 0.8, and 0.1 in the user study — the parameters were tuned through a manual grid search over the discrete set $\{0.0, 0.1, \ldots, 0.9, 1.0\}$ for α and β ($\gamma = 1 - \alpha - \beta$) to obtain the most coherent clusters across all natural disasters. While the weight we used notably places heavy emphasis on distance in the spatial dimension, we note that other values (e.g., balanced weights) led to clusters with more extreme overlapping spatial dimensions that cluttered the user interface, hence justifying the higher weighting for the spatial dimension. Once clusters are extracted by *X*-means, we use the EF1 metric to extract the top clusters as required by our interface.

6. Study 1: Offline evaluation

While our online user study will allow us to evaluate whether our relevance-driven approach to clustering enhances user performance in a visual search task (compared to K-means clustering and a multiple filter search baseline), we first wish to understand how our greedy algorithm compares to other clustering methods (including the optimal MILP solution) as we vary properties of the data and relevance score noise. Because this evaluation requires thousands of independent trials for each possible experimental configuration of data size size, noise level, and label balance level that would have prohibitive time and resource requirements for a user study, we opt to perform these evaluations through offline methods. To perform this offline study, we remark that we optimize clusters via EF1 that is based on the probabilistic relevance scores of the elements obtained at query time ($S_q(j)$), whereas we experimentally evaluate the *actual* quality of the clusters generated using F1 based on the known ground truth ($B_q(j)$), which is standard procedure in an offline evaluation of conventional IR tasks.

Our main objective in this section is to benchmark and intuitively understand the performance of **RadiCAL-Greedy**, **RadiCAL-BPS**, and **K-means** algorithms w.r.t. clusters obtained via the **RadiCAL-MILP** solution (optimizing EF1 using a MILP) through an *offline* empirical evaluation. As an additional control, we also include an evaluation of the **Initial** maximal cluster of all information elements E that is

the initial starting condition for **RadiCAL-Greedy**, **RadiCAL-BPS**, **RadiCAL-MILP, and K-means**. By definition, **Initial** is a trivial recall-maximizing (but low precision) baseline that all algorithms should ideally improve on.

Because we aim to evaluate the impact of noisy relevance evaluations on cluster quality w.r.t. actual ground truth F1-score (*not* the estimated EF1), it is critical to explicitly control noise levels, which we achieve through a noisy corruption of ground truth. Hence, for each tweet j, we assign the probability $S_q(j)$ to that tweet to indicate its relevance probability by introducing a random noise signal as follows: $S_q(j) = \lambda \times B_q(j) + (1 - \lambda) \times \text{rand}()$, where $B_q(j)$ is the boolean ground truth relevance, rand() is a random noise value chosen with uniform distribution in the range $[0, 1]$, and λ is a weighting parameter ($0.5 \leq \lambda \leq 1$) that controls the signal-to-noise ratio in the final probability value. Note that for $\lambda = 1$, $S_q(j)$ is a perfect predictor of ground truth probability, whereas for $\lambda = 0.5$, $S_q(j)$ is extremely noisy (i.e., the signal-to-noise ratio is 1).

We explicitly vary the number of relevant information elements in our ground truth to assess performance variation as a function of class imbalance. In our experimental comparison, we only evaluate **RadiCAL-MILP** up to #data=150 since the MILP solver could not scale to a larger data set. While we do experiment without **RadiCAL-MILP** up to #data=10^4, the smallest positive rate that we can practically evaluate for *all* algorithms in these experiments is 1% given the data size restrictions of **RadiCAL-MILP**.

Each evaluation was carried out by averaging over 10 independent runs that each select random relevant documents according to the designated #data size and positive rate. We report the average ground truth F1-Score (i.e., ground truth is known in the experimental setting) for the EF1-maximizing cluster produced by each method. We report and discuss the main results of the experimental evaluation, considering both the accuracy and the effectiveness of the described algorithms. The configuration options that we have evaluated are the following: F1-Score vs. $\#data \in \{10, \ldots, 150, \ldots, 10^4\} \times \lambda \in \{0.6, 0.7, 0.8, 0.9, 1.0\} \times$ rate of positive data $\in \{1\%, 2\%, 10\%, 50\%\}$.

Varying #data: Here we aim to understand how the different F1-Score optimization algorithms perform as the amount of data varies for differing noise levels. This analysis is provided in Figure 3 while fixing the rate of positive data to 2% and $\lambda \in \{0.6, 0.9, 1.0\}$, and varying the size of the data.

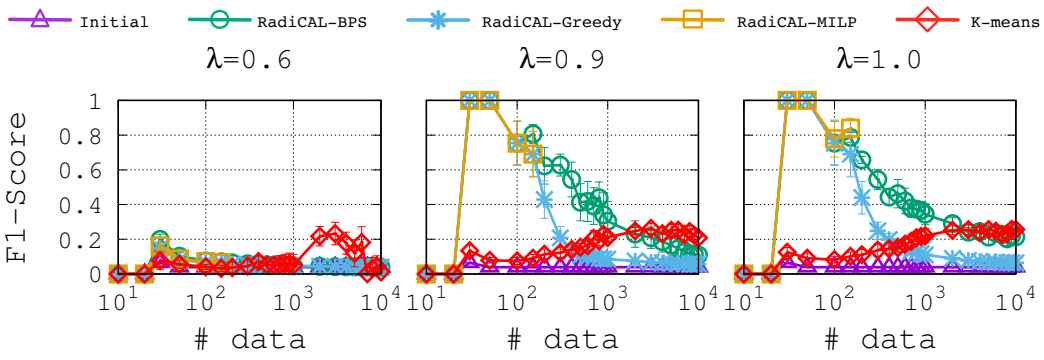


Figure 3: Clustering algorithm performance vs. varying #data for 2% of positive data.

In Figure 3, we observe that beyond a certain #data threshold, the best F1 score decreases since clusters will inevitably include some irrelevant elements in large datasets. Nonetheless, we observe that for relatively low noise ($\lambda = 0.9$ and $\lambda = 1.0$), **RadiCAL-BPS** and **K-Means** perform comparably for large data sizes, while **K-Means** performs poorly for smaller data sizes, but **RadiCAL-BPS** performs better for moderate data sizes and is in fact *provably optimal* (i.e., it overlaps with **RadiCAL-MILP**) for the data sizes that **RadiCAL-MILP** can solve. Quite interestingly, **RadiCAL-BPS** generally outperforms **RadiCAL-Greedy** since the small incremental EF1 improvement steps of **RadiCAL-Greedy** are prone to local optima.

While it may seem concerning that **RadiCAL-BPS** performs poorly for $\lambda = 0.6$, we note that this is an *extremely high level of noise* and thus the relevance estimates are highly unreliable. In this case, a clustering method such as **K-Means** is clearly better off in that it simply ignores the unreliable relevance scores. This leads to the obvious but important conclusion that *the relevance-driven clustering methods proposed in this article should only be used when it is believed that the relevance signal is reasonably reliable*; this *should* be the case in many domains where modern information retrieval scoring techniques are already used.

Varying relevance noise (λ): Here we aim to understand how the different clustering algorithms perform as the amount of noise λ (defined previously) in the relevance prediction varies. The results of this analysis are shown in Figure 4 while having the rate of positive data in {1%,10%,50%}, #data=150 (for comparison to **RadiCAL-MILP**), and varying the amount of noise $\lambda \in [0.6, 1.0]$.

In Figure 4, we observe that for a low rate of positive data the problem becomes easier (higher F1-Score value) as λ increases because EF1 becomes closer to the ground truth F1-Score. Here, **K-Means** performs poorly for higher λ since it ignores the relevance signal, while **RadiCAL-BPS** performs near to **RadiCAL-MILP**; **RadiCAL-Greedy** matches it only in the zero noise case where **RadiCAL-Greedy** cannot get stuck in local optimal.

It is interesting to analyze the boundary case for high rates of positive data (10% and 50%), which is plausible if clustering is applied to the top-ranked search results of an unambigious query. In this case, the relevance noise level has little impact on the F1-score of the algorithms as information elements are more

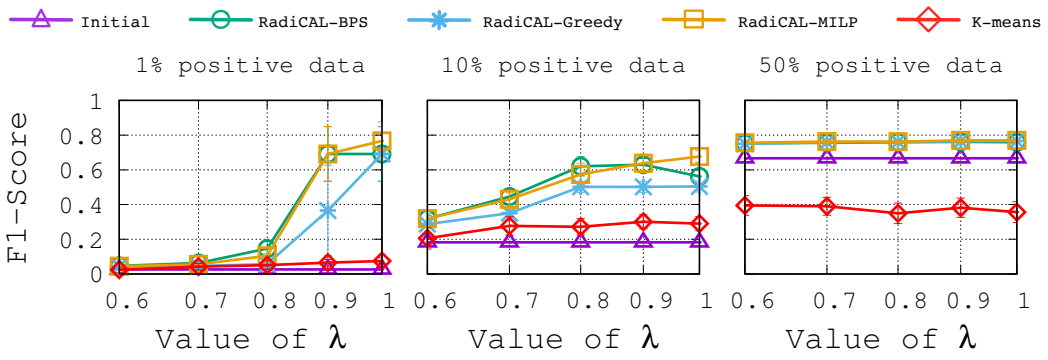


Figure 4: Clustering algorithm performance vs. varying relevance noise (λ) for #data=150.

often relevant than not. Nonetheless, it is telling that **K-Means** does so poorly: it appears to focus on low relevance, but highly self-similar clusters of irrelevant data – a critical caveat of ignoring relevance scores.

In conclusion, the evaluations have shown that the **RadiCAL-Greedy** and **RadiCAL-BPS** algorithms are a good approximation of the Optimal MILP formulation (**RadiCAL-MILP**) and may outperform the **RadiCAL-MILP** in high noise settings — especially the **RadiCAL-BPS** approach which tends to overfit less to noise. While this offline evaluation methodology allows direct head-to-head comparison of clustering algorithm properties and demonstrates potential advantages of **RadiCAL-BPS**, what we need to experiment with next is whether **RadiCAL-BPS** and our relevance-driven clustering approach enhance actual user performance in an online end-to-end visual search task.

7. Study 2: User study

To complement and validate the results of the offline evaluation, we ran a user study with 24 subjects to measure the performance and preference of users with different clustering algorithms for a visual search interface. In the following, we first briefly describe the way we estimate the relevance probability of each tweet in the visual search interface of the user study, then we describe the full user study methodology, and finally we end with an analysis of user performance.

7.1. Relevance Scoring

The underlying search and relevance scoring tool we developed for this user study was built on top of the Lucene IR System[7]. As shown in Figure 5, the interface allows users to enter a multi-term search query q, which retrieves the set E_q of top-ranked 1,000 tweets according to their probability of relevance. As we previously defined in our relevance-driven clustering approach, terms (a.k.a. keywords) can be included/excluded by the clustering algorithm during the cluster extraction process. However, in this user study, terms also

[7]http://lucene.apache.org/

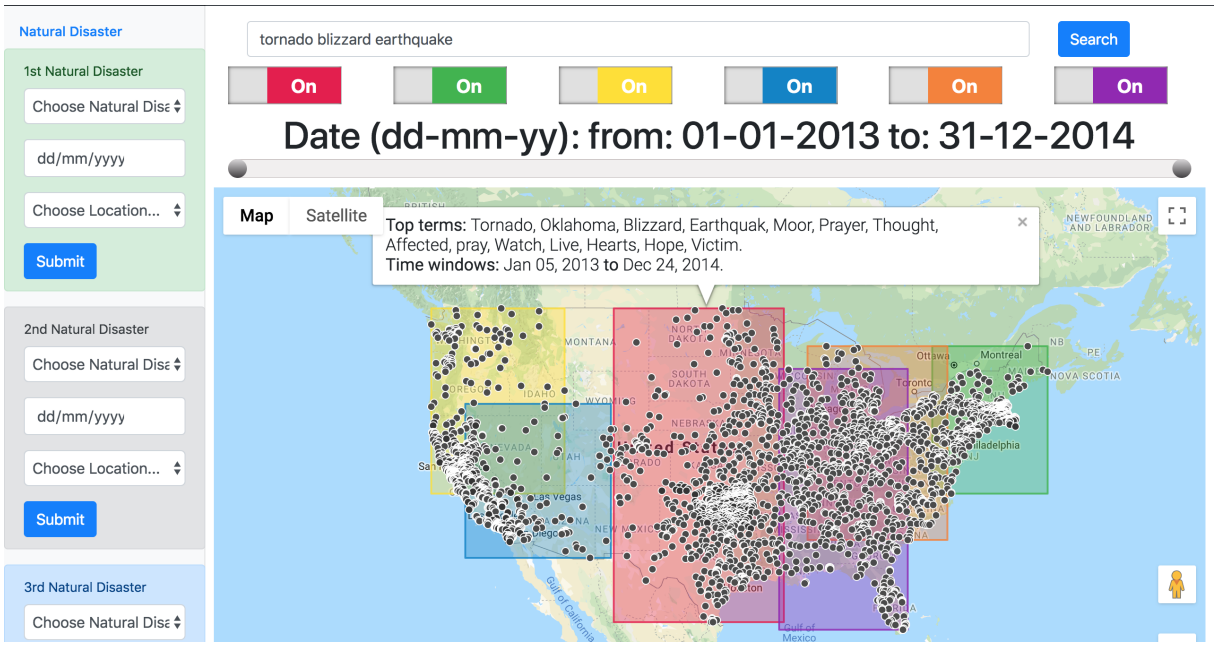


Figure 5: Clustering-based visual search interface. The results of the query "tornado, blizzard, earthquake" are shown using clusters of K-means. The top bar provides both a search box as well as buttons to explicitly hide or show clusters in the map display that allow the user to navigate and analyze clusters even when they overlap (e.g., as occurs with the yellow and blue clusters). The left sidebar is used to provide answers in our experimental user study evaluated in this section.

enable us to derive the probability of relevance of a tweet to the user query. Specifically, we used the standard information retrieval Language Model relevance scoring method as defined in [70] for estimating the relevance score of a tweet j w.r.t. a user query q as follows:

$$S_q(j) = p(q|j) = \prod_{i=1}^{n} p(q_i|j)\,. \tag{8}$$

Here, $p(q_i|j)$ is a unigram language model based on the tweet content j calculated according to a Bayesian smoothing language model using Dirichlet priors as described in Section 3 of [70]. The relevance score provides the probability of relevance for each tweet that is subsequently used to compute and optimize the *expected* F1-Score cluster extraction for our relevance-driven clustering.

7.2. Research Hypotheses and Evaluation Methodology

Designing a user study for Interactive IR (IIR) is a complex and challenging task, but fortunately one for which there is excellent guidance in the literature [77, 78]. Beyond standard randomization protocols that we describe in the following methodology, a key recommendation for IIR evaluation is to construct simulated scenarios and tasks in order to engage participants in the search in a way that is as close as possible to actual information searching and IR processes; this guiding principle is a cornerstone of the interactive visual search task that we have designed in this study. We further remark that we have intentionally designed independent trials with non-dynamic information needs to avoid more complex IIR evaluation considerations, cf. [77].

The main goal of this user study was to comparatively evaluate human performance and visual search interface preference using three different search interfaces for identifying facts about natural disaster data in the previously described Twitter dataset. The primary two hypotheses that we aimed to evaluate with human subjects were the following: (**H1**) a clustering-based interface leads to better search performance and is more preferred than a non-clustering **Baseline** interface, and (**H2**) the relevance- and interface-driven clustering of **RadiCAL-BPS** leads to better search performance and is more preferred by users than **K-means**.

Over a sequence of three trials, each using a randomized (non-overlapping) selection of three natural disasters chosen from Table 1, each of our users was asked to use one of the following three different search approaches (with the order of the three search approaches randomized for the three trials of each user):

1. A **Baseline** multiple filter search method which displays **all results that match the query**. An example of the map portion of the display is shown in Figure 1a with relevance shown as a gray-level shading. While pan and zoom modulate the spatial filter, a time range adjustment filter is provided in addition to a keyword search filter to control inclusion/exclusion of content with specific terms.
2. The **K-means** algorithm discussed previously in the offline evaluation (using X-means to automatically identify the best K), which displays the **largest 6 clusters for results matching a query** (an example is shown in Figure 5).
3. The]**RadiCAL-BPS** algorithm we proposed for relevance-driven clustering that substantially outperformed the **Greedy** approach in the offline experimentation. To match the presentation of **K-means**, **RadiCAL-BPS** displays the **top 6 EF1-scoring clusters for results matching a query** (an example with 3 clusters is shown in Figure 1c).

Overall, we believe that the **K-means** and multiple filter **Baseline** represent ideal methods for comparison to **RadiCAL-BPS** since the first is arguably the most commonly used clustering method used in practice and the latter represents the manually-driven multiple filter search approach that **RadiCAL-BPS** is attempting to automate through optimized relevance-based extraction of clusters defined by filter criteria. Though use of the **Baseline** method would be visually apparent to users, users were not aware of which clustering algorithm they were using in a trial that used either **K-means** or **RadiCAL-BPS** clustering.

For each search approach, the interface allows users to enter a multi-term search query. In the **Baseline** search approach, these tweets are displayed on a Google Maps display used to browse the results. The tweets that match a query were represented using circles with a grayscale color range corresponding to the probability of relevance – light gray circles represented low probability relevance tweets, and dark gray circles represented high probability relevance tweets. The user was able to interact with the map by panning and zooming and also by clicking on tweets to see their content. The clustering search approaches (**K-means** and **RadiCAL-BPS**) were identical to the **Baseline** search approach, except that instead of showing all matching results, they showed six clickable clusters of results as illustrated in Figure 5 (clicking a cluster displays a summary view and clicking a tweet in the cluster displays the specific tweet content). In all search approaches, the user could use a time slider bar to restrict the results to tweets matching the query in a

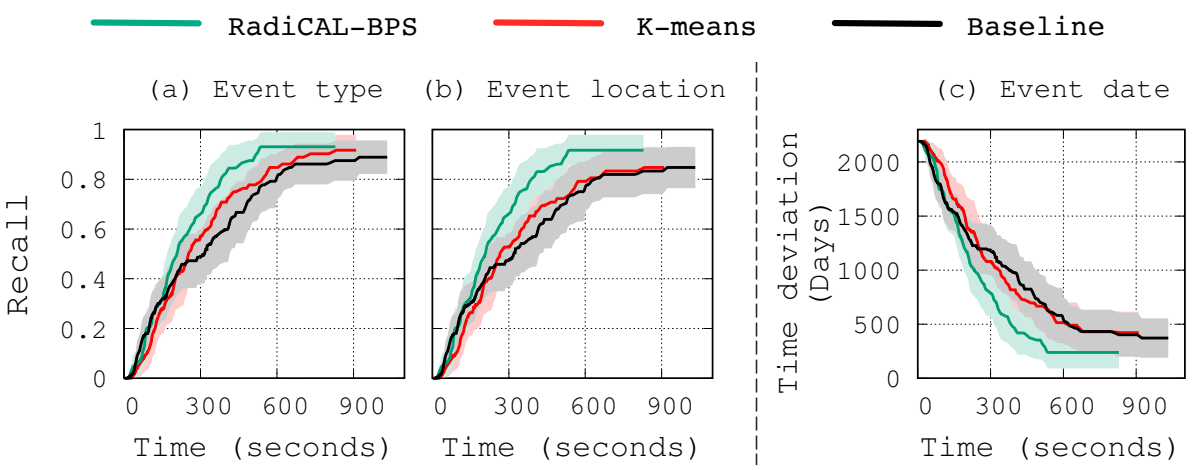


Figure 6: The mean user performance and 95% confidence intervals for each of the three search interfaces/algorithms are measured using cumulative recall for the type and location of the natural disasters and the absolute error for the first date of the natural disaster. On average, users achieved higher recall and lower error faster using relevance-driven clustering (**RadiCAL-BPS**) in comparison to K-means and the Baseline.

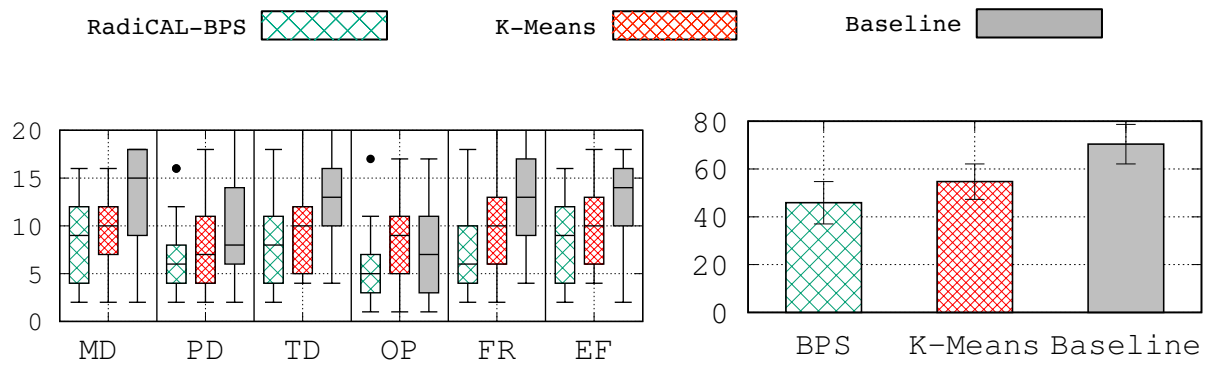
specific time window.

Before starting the experiment, each user was provided with simple instructions to turn off all personal devices for the duration of the experiment and then shown a training video describing the visual search interface and how to interact with the clusters (i.e., how they could change their search queries, pan/zoom, use the time slider, and enter their answers). Then, each user was tested on their ability to find each of three natural disasters from Table 1 using both the **Baseline** non-clustering interface as well as the **K-means** clustering interface in two training trials. In each of the two training trials and the three experimental trials, the user was asked to enter information related to each natural disaster they identified, including the type of the natural disaster (e.g., earthquake, hurricane, flood, etc.) selected from a drop-down list, its location (US state) selected from a drop-down list, and the date (day) on which they think the disaster first occurred selected from a calendar chooser; the area where this information is entered can be seen on the left-hand bar in Figure 5. It is important to reiterate that *none* of the three experimental trials reused data from a previous trial. A total of 24 users participated in the user study, for which the full experiment took on average 50 minutes per user.

We collected detailed interaction logs to record different behaviors and actions of the user. Moreover, each user was asked to answer NASA Task Load Index (NASA-TLX) [79] and System Usability Scale (SUS) [80] questionnaires after each trial. Finally, at the end of the experiment, each user was asked to fill out an exit survey, which included a preference ranking of the algorithms.

7.3. Quantitative performance analysis

The performance of the users for each algorithm was measured using cumulative recall for the type and location of the natural disasters in each trial. Since there were three distinct natural disasters per trial, perfect



(a) Subscales rating.

(b) Overall rating.

Figure 7: NASA Task Load Index (MD: Mental Demand, PD: Physical Demand, TD: Temporal Demand, OP: Own Performance, FR: Frustration, EF: Effort). Higher numbers indicate higher perceived workload.

recall would require getting all three natural disaster types or locations correct. Mean performance across all users with 95% confidence intervals for cumulative recall of disaster type and location are respectively shown in Figures 6(a) and 6(b). Users also had to enter their estimated starting date of each natural disaster, which we measure by absolute error assuming the maximum error for natural disasters that have not been submitted yet. The mean performance of time estimation error across all users with 95% confidence intervals is reported in Figure 6(c).

Regarding the quantitative portion of hypotheses H1 and H2, there are very consistent user performance trends that emerge after 150 seconds into the trial in Figure 6. In the first 150 seconds, it would appear that all users were adjusting to the given task and some users were able to identify some of the most salient natural disasters that would have been obvious regardless of the interface type. It seems that it is beyond this initial stage when users are searching for the remaining less salient natural disasters when the interface helps differentiate human performance. Specifically, after 150 seconds in Figure 6, *we observe the general trend that, on average, users achieved higher task recall and lower error faster* when using the **RadiCAL-BPS** algorithm for clustering in comparison to **K-means** and the **Baseline**. Furthermore, we clearly notice that with the **RadiCAL-BPS** algorithm, participants were able to achieve a *higher average "asymptotic" performance at an earlier stage of the experiment* than using the two other search algorithms. Such results quantitatively support both of our experimental hypotheses H1 and H2 from Section 7.2 that motivated our study.

7.4. Survey analysis

With the previous quantitative measures of user performance indicating the advantage of relevance-driven clustering, we next proceed to evaluate the users' own opinions of each interface/algorithm as collected in the user surveys discussed in the methodology. During the user study, each participant answered 7 different questionnaires – a NASA Task Load Index (NASA-TLX) [79] and a System Usability Scale (SUS) [80]

questionnaire after using each algorithm, plus a final questionnaire. We report key results below.

7.4.1. NASA-TLX analysis

The NASA-TLX questionnaire rates perceived workload in order to assess a task, system effectiveness or other aspects of performance. The questionnaire includes questions on mental demand (MD), physical demand (PD), temporal demand (TD), performance (OP), frustration (FR), and effort (EF). NASA-TLX items are rated on a 20-point scale (1 = low workload, 20 = high workload, except for OP where 1 = perfect and 20 = failure). Overall, we hypothesize that the three most important factors for this visual search task are temporal demand (reduced time to complete the search task owing to better clusters), mental demand (the ability to focus analysis at the cluster level of abstraction as opposed to the tweet level), and effort (reduced effort to analyze clusters due to clear keyword summaries). We conjecture that all of these task load reductions would follow from the increased coherence of the **RadiCAL-BPS** relevance-driven cluster extraction compared to unsupervised **K-means** clustering and the lack of any automatic clustering in the **Baseline**.

We show the NASA-TLX results obtained (a) for each subscale and (b) for the overall ratings in Figure 7. Briefly, the mean overall NASA-TLX rating was 45.91 ± 8.86 for **RadiCAL-BPS**, 54.75 ± 7.41 for **K-means**, and 70.41 ± 8.27 for the **Baseline**. A Friedman's test revealed an overall significant difference ($\chi^2(3) = 16.113, p = 0.003 < 0.05$). Holm-Bonferroni corrected post-hoc analyses with Wilcoxon signed-rank tests revealed that the difference between all pairs was significant ($p < 0.05$).

We note that participants overall perceived the **RadiCAL-BPS** algorithm to be more effective at helping them complete their search task in comparison to using **K-means** or the **Baseline**. Considering each of the three aforementioned key factors (TD, MD, EF) deemed most relevant to the innovations of the **RadiCAL-BPS** clustering algorithm, we remark that **RadiCAL-BPS** recorded the lowest median load on all three factors with K**-means** somewhat behind in second place (indicating that some form of clustering still aided the visual search task) and the non-clustering **Baseline** further afield with the highest loads. Considering additional factors, global median rates of frustration and mental effort were around 10, which indicates that the task was neither too difficult, nor too easy. Also, as the median rates of these two factors were lower for the two clustering methods than the baseline, we conclude that participants overall felt that clustering-based search provided a less frustrating and less mentally demanding interface for this task in comparison to the baseline, which displayed all search results. Finally, based on the task load rates where the **RadiCAL-BPS** algorithm performed the best, it seems apparent that relevance-driven **RadiCAL-BPS** clustering provided the most effective approach for carrying out this kind of spatio-temporal search task, which is supported by the overall rating of Figure 7(b).

7.4.2. SUS analysis

The SUS questionnaire gives a global view of subjective assessments of usability. The questionnaire contains questions related to the global effectiveness, efficiency, and satisfaction. The SUS items are rated

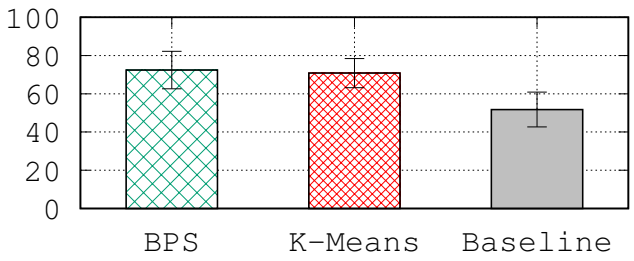


Figure 8: System Usability Scale overall rating. Higher values indicate higher perceived usability.

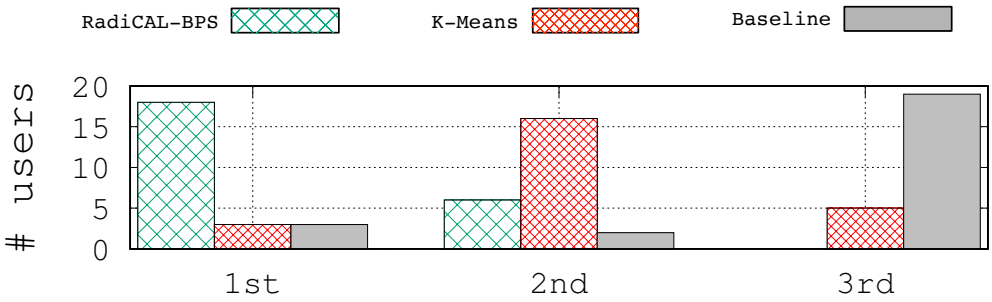


Figure 9: Ranking of the algorithm preference by participants who voted by trial ID (not algorithm name). The number of users who placed each algorithm at the specified ranking is shown. Clearly, relevance-driven clustering (**RadiCAL-BPS**) received the bulk of first place preference, K-means the bulk of second place preference, and Baseline the least preference at third place.

on a 5-point scale (0 = strong disagreement and 5 = strong agreement). In Figure 8, we show the results obtained over all SUS scores with 95% confidence interval. The mean SUS score was 72.39 ± 9.77 for **RadiCAL-BPS**, 70.83 ± 7.61 for **K-means**, and 51.77 ± 9.07 for the **Baseline**. A Friedman's test revealed an overall significant difference ($\chi^2(3) = 9.053, p = 0.010 < 0.05$). Holm-Bonferroni corrected post hoc analyses with Wilcoxon signed-rank tests revealed that the difference between the two clustering methods and the baseline was significant ($p < 0.05$). The difference between **RadiCAL-BPS** and **K-means** wasn't significant and hence we can only infer from the SUS survey that the users preferred the clustering interface (i.e., **RadiCAL-BPS** and **K-means**) over the **Baseline**.

7.4.3. Final questionnaire analysis

The final questionnaire that the users had to answer included questions on the advantages and disadvantages of each algorithm, plus a global ranking of the three algorithms. The ranking of the algorithms provided by users is shown in Figure 9. We note that 18 users out of 24 ranked **RadiCAL-BPS** as being the best algorithm, and 19 users out of 24 ranked the **Baseline** approach as being the least helpful for the task. Hence there is strong evidence for user preference of **RadiCAL-BPS** over **K-means** and for both **RadiCAL-BPS** and **K-means** over the **Baseline**. In the free response survey section, several users also specifically reported the ease and precision provided by the trial with the **RadiCAL-BPS** algorithm assigned, while no users indicated this for the trials with **K-means** or **RadiCAL-BPS** assigned.

Combined with the quantitative performance analysis of Figure 6 discussed in Section 7.3 and the NASA-TLX workload and SUS usability survey results discussed in Section 7.4 that both corroborate these preference findings, we remark that all of the experimental evidence strongly supports hypotheses H1 and H2 from Section 7.2 that motivated our user study.

8. Discussion

In this section, we discuss a summary of our key contributions and results, the main limitations of this research, and possible directions for future work.

8.1. Summary

In this article, we began by observing that unsupervised clustering methods have often been used to aggregate data in visual search interfaces, but approaches like K-means do not make effective use of query relevance signals during this aggregation task and do not necessarily optimize for purposes of visual presentation in the user interface. To address these deficiencies, we introduced novel and efficient relevance-driven clustering approximate optimization algorithms for expected F1-Score based on two different greedy strategies (**RadiCAL-Greedy** and **RadiCAL-BPS**).

The offline evaluations we performed show that the binary partitioning search (**RadiCAL-BPS**) algorithm we have proposed is relatively efficient, performs comparably to or exceeds K-means performance when the relevance signal is moderately reliable, and provides a good approximation of the *optimal* MILP solution in small instances where comparison is possible.

The user study we carried out on 24 users confirmed the outcome of the offline evaluation and has demonstrated that our novel relevance-driven clustering based on BPS (**RadiCAL-BPS**) is highly effective for our search scenario. Specifically we confirmed quantitatively that users achieved generally faster search task completion, higher recall, and lower error using relevance-driven clustering compared to K-means and a non-aggregation baseline. Users also indicated that the **RadiCAL-BPS** approach yielded lower perceived workload on the NASA-TLX survey and higher usability on the SUS survey. And finally, to corroborate all of these findings, users ultimately indicated a strong first preference for the relevance and interface-driven clustering approach that comprises the novel contribution of this article. All the algorithms described throughout this paper have been integrated into a tool called Visual Twitter Information Retrieval (Viz-TIR) [23].[8]

8.2. Limitations

While we believe this work has made a number of significant contributions to both visual search interfaces as well as the novel area of relevance-driven clustering, it is critical to acknowledge the following potential limitations of the present work:

[8] `https://github.com/D3Mlab/viz-ir`

- *The limitation of a pure relevance-driven clustering approach for visual search interfaces.* While we believe that clustering algorithms for visual search interfaces should include relevance as part of their clustering criteria, it is likely that there are additional qualities of a "good" cluster for end users in visual search interfaces (e.g., dense clusters) that could be folded into an enhanced clustering objective. Introducing such additional criteria in the objectives inherently raise multiobjective optimization concerns regarding optimal trade-offs among all objective criteria [81].

- *Unintended study participant variation due to lack of context regarding natural disasters in the US.* We chose US natural disasters due to the prevalence of tweet content on this topic. However, we remark that we ran our study with University students residing in Toronto, Canada. Even though instructions and suggestions were given to the participants regarding the possible natural disasters to search, some participants mentioned the difficulty to come up with the right queries, mostly because of their unfamiliarity with the US context.

- *The limited number of users.* We ran our user study with the maximum number of users that we could run under time and budget constraints. For this reason, we could not compare a large number of algorithms and furthermore we were unable to draw conclusions in our user study with a high degree of statistical significance. Hence, we believe that our user study and its conclusions should be reinforced with more participants as well as more clustering algorithms to be compared.

- *Limitation of use cases to low noise queries.* One key limitation of our relevance-driven clustering algorithm was evidenced in our offline study of Section 6 when we noted that high noise in the relevance score led to K-Means clustering outperforming the RadiCAL variants since in this case, ignoring the relevance signal was advantageous. For this reason, we chose natural disasters in our dataset curation that are concrete events whose associated queries tend to have relatively low noise in their relevance scores. However, visual search use cases where queries are difficult to formulate and relevance harder to predict (e.g., searches for public sentiment on political topics) may prove difficult for the relevance-driven clustering techniques proposed here and require alternatives or enhancements to work well.

- *Adding more clustering dimensions to the algorithm may not be trivial.* Although we mentioned previously that we are not limited to the three display attributes of space, time, and keyword content, it is not immediately clear how our efficient greedy approximation algorithms will perform for additional meta-data dimensions. In particular, such dimensions may violate the property that the clustering objective changes relatively monotonically as each cluster dimension bound is adjusted, which was an assumption underlying the application of the highly efficient BPS approach.

8.3. Future Work

Overall, we believe this work underscores the importance of relevance-driven clustering optimization methods specifically targeted for presentation in interactive visual search interfaces. However, there is still

much more work to be done to fully explore the space of optimization objectives and algorithms for interactive visual search. Interesting areas of future work include the following:

- *Augmenting the relevance-focused F1-Score with additional cluster criteria to enhance word-level, spatial, and temporal coherence.* While we argued and empirically demonstrated that our expected F1-Score cluster optimization objective does lead to coherently interpretable clusters, there is certainly the possibility to explore more complex objectives that may further enhance coherency for end users. For example, research could explore novel application-specific objectives that take into account physical constraints of the display device as well as user cognitive constraints and preferences for cluster display that could be used to augment (or even replace) the existing F1-Score relevance-based objective.

- *Considering a ranking perspective of cluster relevance and optimization as opposed to a Boolean perspective.* In this initial work, we did not want to address ranking aspects of visual clustering that might entail showing each result in a cluster as a different size since this would require consideration of psychovisual aspects of human information processing that are beyond the scope of the present article. Nonetheless, such an extension is certainly possible in future work and would better leverage additional degrees of freedom in the search results display to distinguish results by their level of relevance. A key challenge in such an extension would involve developing new optimization algorithms that could deal with the increased complexity of a ranking-style evaluation metric of cluster relevance.

- *Considering a topic modeling perspective in place of a clustering perspective.* While clustering attempts to aggregate similar documents with the assumption that each document belongs to one cluster, topic modeling methods such as (probabilistic) LSA [82, 83] or LDA [84] assume that a document is composed of multiple topics and try to estimate the degree or probability of relevance that a document has to each topic. While topic modeling provides a more granular view of document content, such approaches pose a number of significant challenges for use with visual search interfaces. Specifically, this would require highly effective spatio-temporal extensions of topic models, novel methods to visually display topics as opposed to clusters, extensions to topic modeling that explicitly consider relevance-based optimization criteria, and novel computational methods to effectively optimize within this extended framework. Nonetheless, the benefits of a more granular topic modeling perspective may certainly motivate extensions of this work to *relevance-driven topic modeling* for visual search interfaces.

In sum, we hope that this article provides a first stepping stone to a wide range of exciting future work on the topic of relevance- and interface-driven clustering to help create the next generation of enhanced visual information retrieval systems.

Acknowledgements

We thank the two anonymous reviewers for their questions and comments, which helped to improve and clarify the article presentation and also provided numerous suggestions for our future work discus-

sion. All user experiments were conducted under the approved University of Toronto RIS Human Protocol #36140. Experiments presented in this paper were carried out using the NECTAR Research Cloud platform, supported by a scientific interest group hosted by several Australian Universities (see `https://nectar.org.au/research-cloud/`). All algorithms described in this paper, the material used, as well as the tool we developed for the user study can be accessed here: `https://github.com/D3Mlab/viz-ir`.

References

[1] Benjamin E. Teitler, Michael D. Lieberman, Daniele Panozzo, Jagan Sankaranarayanan, Hanan Samet, and Jon Sperling. Newsstand: A new view on news. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '08, pages 18:1–18:10, New York, NY, USA, 2008. ACM.

[2] Jagan Sankaranarayanan, Hanan Samet, Benjamin E. Teitler, Michael D. Lieberman, and Jon Sperling. Twitterstand: News in tweets. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, pages 42–51, New York, NY, USA, 2009. ACM.

[3] Amr Magdy, Louai Alarabi, Saif Al-Harthi, Mashaal Musleh, Thanaa M. Ghanem, Sohaib Ghani, and Mohamed F. Mokbel. Taghreed: A system for querying, analyzing, and visualizing geotagged microblogs. In *Proceedings of the 22Nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '14, pages 163–172, New York, NY, USA, 2014. ACM.

[4] Thanaa M. Ghanem, Amr Magdy, Mashaal Musleh, Sohaib Ghani, and Mohamed F. Mokbel. Viscat: Spatio-temporal visualization and aggregation of categorical attributes in twitter data. In *Proceedings of the 22Nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '14, pages 537–540, New York, NY, USA, 2014. ACM.

[5] Natalia Andrienko, Gennady Andrienko, and Salvatore Rinzivillo. Leveraging spatial abstraction in traffic analysis and forecasting with visual analytics. *Information Systems*, 57:172 – 194, 2016.

[6] A. Eldawy, M. F. Mokbel, and C. Jonathan. Hadoopviz: A mapreduce framework for extensible visualization of big spatial data. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 601–612, May 2016.

[7] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J.J. van Wijk, J.-D. Fekete, and D.W. Fellner. Visual analysis of large graphs: State-of-the-art and future research challenges. *Computer Graphics Forum*, 30(6):1719–1749, 2011.

[8] Shixia Liu, Weiwei Cui, Yingcai Wu, and Mengchen Liu. A survey on information visualization: recent advances and challenges. *The Visual Computer*, 30(12):1373–1393, Dec 2014.

[9] Guo-Dao Sun, Ying-Cai Wu, Rong-Hua Liang, and Shi-Xia Liu. A survey of visual analytics techniques and applications: State-of-the-art research and future challenges. *Journal of Computer Science and Technology*, 28(5):852–867, Sep 2013.

[10] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[11] Gerard Salton. Cluster search strategies and the optimization of retrieval effectiveness. *The SMART retrieval system-experiments in automatic document processing*, pages 223–242, 1971.

[12] N. Jardine and C.J. van Rijsbergen. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 7(5):217 – 240, 1971.

[13] Ellen M. Voorhees. The cluster hypothesis revisited. In *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '85, pages 188–196, New York, NY, USA, 1985. ACM.

[14] Daniel Tunkelang. Faceted search. *Synthesis lectures on information concepts, retrieval, and services*, 1(1):1–80, 2009.

[15] Jennifer English, Marti Hearst, Rashmi Sinha, Kirsten Medhurst, and Ka-Ping Yee. Flexible search and navigation using faceted metadata. Technical report, University of California, Berkeley, March 2002.

[16] Marti A. Hearst. *Search User Interfaces*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[17] Marti A. Hearst. Clustering versus faceted categories for information exploration. *Commun. ACM*, 49(4):59–61, April 2006.

[18] Christopher Ahlberg and Ben Shneiderman. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In Ronald M. Baecker, Jonathan Grudin, William A.S. Buxton, and Saul Greenberg, editors, *Readings in Human-Computer Interaction (Second Edition)*, Interactive Technologies, pages 450 – 456. Morgan Kaufmann, second edition edition, 1995.

[19] A. Bennamane, H. Hacid, A. Ansiaux, and A. Cagnati. Vizpicious: A visual user-adaptive tool for communication logs analysis and suspicious behavior detection. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 641–642, Dec 2012.

[20] Ben Shneiderman and Cody Dunne. Interactive network exploration to derive insights: Filtering, clustering, grouping, and simplification. In Walter Didimo and Maurizio Patrignani, editors, *Graph Drawing*, pages 2–18, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[21] Hu Yifan and Shi Lei. Visualizing large graphs. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(2):115–136, 2015.

[22] Marc A. Smith, Ben Shneiderman, Natasa Milic-Frayling, Eduarda Mendes Rodrigues, Vladimir Barash, Cody Dunne, Tony Capone, Adam Perer, and Eric Gleave. Analyzing (social media) networks with nodexl. In *Proceedings of the Fourth International Conference on Communities and Technologies*, C&T '09, pages 255–264, New York, NY, USA, 2009. ACM.

[23] Mohamed Reda Bouadjenek and Scott Sanner. Relevance-driven clustering for visual information retrieval on twitter. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, CHIIR '19, pages 349–353, New York, NY, USA, 2019. ACM.

[24] Slava Kisilevich, Florian Mansmann, Mirco Nanni, and Salvatore Rinzivillo. *Spatio-temporal clustering*, pages 855–874. Springer US, Boston, MA, 2010.

[25] Gowtham Atluri, Anuj Karpatne, and Vipin Kumar. Spatio-temporal data mining: A survey of problems and methods. *ACM Comput. Surv.*, 51(4):83:1–83:41, August 2018.

[26] E. Eftelioglu, S. Shekhar, D. Oliver, X. Zhou, M. R. Evans, Y. Xie, J. M. Kang, R. Laubscher, and C. Farah. Ring-shaped hotspot detection: A summary of results. In *2014 IEEE International Conference on Data Mining*, pages 815–820, Dec 2014.

[27] Hamed Abdelhaq, Christian Sengstock, and Michael Gertz. Eventweet: Online localized event detection from twitter. *Proc. VLDB Endow.*, 6(12):1326–1329, August 2013.

[28] Flavio Chierichetti, Jon M Kleinberg, Ravi Kumar, Mohammad Mahdian, and Sandeep Pandey. Event detection via communication pattern analysis. In *ICWSM*, 2014.

[29] Maximilian Walther and Michael Kaisser. Geo-spatial event detection in the twitter stream. In *Advances in Information Retrieval*, pages 356–367, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[30] J. Chae, D. Thom, H. Bosch, Y. Jang, R. Maciejewski, D. S. Ebert, and T. Ertl. Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 143–152, Oct 2012.

[31] Yan-Tao Zheng, Zheng-Jun Zha, and Tat-Seng Chua. Mining travel patterns from geotagged photos. *ACM Trans. Intell. Syst. Technol.*, 3(3):56:1–56:18, May 2012.

[32] Ke Xie, Chaolun Xia, Nir Grinberg, Raz Schwartz, and Mor Naaman. Robust detection of hyper-local events from geotagged social media data. In *Proceedings of the Thirteenth International Workshop on Multimedia Data Mining*, MDMKDD '13, pages 2:1–2:9, New York, NY, USA, 2013. ACM.

[33] Aharona Glatman-Freedman, Zalman Kaufman, Eran Kopel, Ravit Bassal, Diana Taran, Lea Valinsky, Vered Agmon, Manor Shpriz, Dani Cohen, Emilia Anis, and Tamy Shohat. Near real-time space-time cluster analysis for detection of enteric disease outbreaks in a community setting. *The Journal of infection*, 73, 06 2016.

[34] Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28:129–137, 1982.

[35] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, SIGMOD '96, pages 103–114, New York, NY, USA, 1996. ACM.

[36] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, SIGMOD '99, pages 49–60, New York, NY, USA, 1999. ACM.

[37] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press, 1996.

[38] Derya Birant and Alp Kut. St-dbscan: An algorithm for clustering spatial-temporal data. *Data & Knowledge Engineering*, 60(1):208 – 221, 2007. Intelligent Data Mining.

[39] Dong-Wan Choi and Chin-Wan Chung. A k-partitioning algorithm for clustering large-scale spatio-textual data. *Information Systems*, 64:1 – 11, 2017.

[40] Natasa Tagasovska and Periklis Andritsos. Distributed clustering of categorical data using the information bottleneck framework. *Information Systems*, 72:161 – 178, 2017.

[41] Saeed Shahrivari and Saeed Jalili. Single-pass and linear-time k-means clustering based on mapreduce. *Information Systems*, 60:1 – 12, 2016.

[42] Hai-Tao Yu, Adam Jatowt, Roi Blanco, Hideo Joho, Joemon M. Jose, Long Chen, and Fajie Yuan. Revisiting the cluster-based paradigm for implicit search result diversification. *Information Processing & Management*, 54(4):507 – 528, 2018.

[43] Chunlin Li, Jingpan Bai, Zhao Wenjun, and Yang Xihao. Community detection using hierarchical clustering based on edge-weighted similarity in cloud environment. *Information Processing & Management*, 56(1):91 – 109, 2019.

[44] Qi-Zhu Dai, Zhong-Yang Xiong, Jiang Xie, Xiao-Xia Wang, Yu-Fang Zhang, and Jia-Xing Shang. A novel clustering algorithm based on the natural reverse nearest neighbor structure. *Information Systems*, 84:1 – 16, 2019.

[45] Lili Kotlerman, Ido Dagan, and Oren Kurland. Clustering small-sized collections of short texts. *Inf. Retr. Journal*, 21(4):273–306, 2018.

[46] Or Levi, Ido Guy, Fiana Raiber, and Oren Kurland. Selective cluster presentation on the search results page. *ACM Trans. Inf. Syst.*, 36(3):28:1–28:42, February 2018.

[47] Ismail Sengor Altingovde, Engin Demir, Fazli Can, and Özgür Ulusoy. Incremental cluster-based retrieval using compressed cluster-skipping inverted files. *ACM Trans. Inf. Syst.*, 26(3):15:1–15:36, June 2008.

[48] Fazli Can, Ismail Sengör Altingövde, and Engin Demir. Efficiency and effectiveness of query processing in cluster-based retrieval. *Inf. Syst.*, 29(8):697–717, December 2004.

[49] Hiroyuki Toda and Ryoji Kataoka. A search result clustering method using informatively named entities. In *Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management*, WIDM '05, pages 81–86, New York, NY, USA, 2005. ACM.

[50] Oren Kurland and Eyal Krikon. The opposite of smoothing: a language model approach to ranking query-specific document clusters. *Journal of Artificial Intelligence Research*, 41:367–395, 2011.

[51] Oren Kurland. Re-ranking search results using language models of query-specific clusters. *Inf. Retr.*, 12(4):437–460, August 2009.

[52] Kathleen R. McKeown, Regina Barzilay, David Evans, Vasileios Hatzivassiloglou, Judith L. Klavans, Ani Nenkova, Carl Sable, Barry Schiffman, and Sergey Sigelman. Tracking and summarizing news on a daily basis with Columbia's Newsblaster. In *Proc. Human Language Technology Conference*, 2002.

[53] Vasileios Hatzivassiloglou, Luis Gravano, and Ankineedu Maganti. An investigation of linguistic features and clustering algorithms for topical document clustering. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 224–231, New York, NY, USA, 2000. ACM.

[54] Xiaoyong Liu and W. Bruce Croft. Cluster-based retrieval using language models. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 186–193, New York, NY, USA, 2004. ACM.

[55] Ismail Sengor Altingovde, Rifat Ozcan, Huseyin Cagdas Ocalan, Fazli Can, and Özgür Ulusoy. Large-scale cluster-based retrieval experiments on turkish texts. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 891–892, New York, NY, USA, 2007. ACM.

[56] Rani Qumsiyeh and Yiu-Kai Ng. Clustering retrieved web documents to speed up web searches. In Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Marina L. Gavrilova, Ana Maria Alves Coutinho Rocha, Carmelo Torre, David Taniar, and Bernady O. Apduhan, editors, *Computational Science and Its Applications – ICCSA 2015*, pages 472–488, Cham, 2015. Springer International Publishing.

[57] Jonathan Dimond and Peter Sanders. Faster exact search using document clustering. In Costas Iliopoulos, Simon Puglisi, and Emine Yilmaz, editors, *String Processing and Information Retrieval*, pages 1–12, Cham, 2015. Springer International Publishing.

[58] Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '92, pages 318–329, New York, NY, USA, 1992. ACM.

[59] Peter L. T. Pirolli. *Information Foraging Theory: Adaptive Interaction With Information.* Oxford University Press, 2007.

[60] Nicholas J. Belkin and W. Bruce Croft. Information filtering and information retrieval: Two sides of the same coin? *Commun. ACM*, 35(12):29–38, December 1992.

[61] Degi Young and Ben Shneiderman. A graphical filter/flow representation of boolean queries: A prototype implementation and evaluation. *J. Am. Soc. Inf. Sci.*, 44(6):327–339, July 1993.

[62] A. Nocaj and U. Brandes. Organizing search results with a reference map. *IEEE Transactions on Visualization & Computer Graphics*, 18:2546–2555, 12 2012.

[63] Shixia Liu, Michelle X. Zhou, Shimei Pan, Yangqiu Song, Weihong Qian, Weijia Cai, and Xiaoxiao Lian. Tiara: Interactive, topic-based visual text summarization and analysis. *ACM Trans. Intell. Syst. Technol.*, 3(2):25:1–25:28, February 2012.

[64] Shixia Liu, Michelle X. Zhou, Shimei Pan, Weihong Qian, Weijia Cai, and Xiaoxiao Lian. Interactive, topic-based visual text summarization and analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 543–552, New York, NY, USA, 2009. ACM.

[65] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. *PLOS ONE*, 9(6):1–12, 06 2014.

[66] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125, Feb 1981.

[67] Tomihisa Kamada and Satoru Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7 – 15, 1989.

[68] Jun Wang and Jianhan Zhu. On statistical analysis and optimization of information retrieval effectiveness metrics. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 226–233, 2010.

[69] Ricardo A Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 2 edition, 2010.

[70] John Lafferty and Chengxiang Zhai. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 334–342, New York, NY, USA, 2001. ACM.

[71] Cyril Goutte and Eric Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In David E. Losada and Juan M. Fernández-Luna, editors, *Advances in Information Retrieval*, pages 345–359, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[72] G.M.P. van Kempen and L.J. van Vliet. Mean and variance of ratio estimators used in fluorescence ratio imaging. *Cytometry*, 39(4):300–305, 2000.

[73] Zahra Iman, Scott Sanner, Mohamed Reda Bouadjenek, and Lexing Xie. A longitudinal study of topic classification on twitter. In *Proceedings of the 11th International AAAI Conference on Web and Social Media a (ICWSM-17)*, pages 552–555, 2017.

[74] A. Charnes and W. W. Cooper. Programming with linear fractional functionals. *Naval Research Logistics Quarterly*, 9(3-4):181–186, 1962.

[75] Fred Glover. Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460, 1975.

[76] Dan Pelleg and Andrew W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 727–734, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[77] Pia Borlund. The iir evaluation model: a framework for evaluation of interactive information retrieval systems. *Information research*, 8(3):8–3, 2003.

[78] Diane Kelly. Methods for evaluating interactive information retrieval systems with users. *Foundations and Trends in Information Retrieval*, 3(1-2):1–224, 2009.

[79] Sandra G. Hart and Lowell E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In Peter A. Hancock and Najmedin Meshkati, editors, *Human Mental Workload*, volume 52 of *Advances in Psychology*, pages 139 – 183. North-Holland, 1988.

[80] John Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[81] Kalyanmoy Deb. Multi-objective optimization. In Edmund K. Burke and Graham Kendall, editors, *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, pages 403–449. Springer US, Boston, MA, 2014.

[82] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R.A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41*, pages 391–407, 1990.

[83] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

[84] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

Appendix A. Optimal MILP Solutions for Benchmarking

In this Appendix, we outline the details of an exact Mixed Integer Linear Programming (MILP) optimization-based formulation to maximize EF1 that provides a benchmark for evaluating the two proposed relevance-driven algorithms (**RadiCAL-Greedy** and **RadiCAL-BPS**) proposed in Section 4 intended to approximately optimize EF1.

Appendix A.1. Fractional MILP Formulation

Leveraging notation defined in Section 3, we begin by reformulating the EF1 objective to prepare for further optimization steps by replacing the global sum of scores of all information elements with a constant $C = \sum_{j=1}^{m} S_q(j)$:

$$EF1 = \frac{2 \times \sum_{j=1}^{m} S_q(j) I_q(j)}{\sum_{j=1}^{m} I_q(j) + \sum_{j=1}^{m} S_q(j)} = \frac{2 \times \sum_{j=1}^{m} S_q(j) I_q(j)}{\sum_{j=1}^{m} I_q(j) + C} \tag{A.1}$$

In order to obtain the EF1-optimal cluster, we let binary variables $I_{filter}(\mathrm{j}) \in \{0, 1\}$ indicate whether an information element j is selected in by each cluster parameter and constrain that to be selected in the Global cluster (i.e., $I_q(j) = 1$), j must be selected by all cluster parameters (i.e., a conjunction). This leads to the following fractional MILP formulation with cluster parameter constraints to be defined later:

$$\begin{aligned} \underset{I_{cluster}(j)}{\text{maximize}} \quad & \frac{\sum_{j=1}^{m} S_q(j) I_q(j)}{\sum_{j=1}^{m} I_q(j) + C} \\ s.t \quad & I_q(j) = \bigwedge I_{cluster}(j) \end{aligned} \tag{A.2}$$

Appendix A.2. Transformation to a MILP

While there are no direct solvers for fractional MILPs, we can transform (A.2) into a pure MILP form for which we have efficient and optimal solvers. To do this, we use the Charnes-Cooper method [74] and Glover linearization method [75] with big-M constraints, where auxiliary variables *w(j)* and u are introduced[9]. Here, $w(j)$ is defined as $w(j) = I_q(j) \times u$ with u defined as follows:

$$u = \frac{1}{\sum_{j=1}^{m} I_q(j) + C} \tag{A.3}$$

Then, the EF1 optimization problem is able to be transformed into the following MILP problem:

$$\begin{aligned} \underset{w,u}{\text{maximize}} \quad & \sum_{j=1}^{m} S_q(j) w(j) \\ s.t \quad & \sum_{j=1}^{m} w(j) + uC = 1 \\ & w(j) \leqslant u, \quad w(j) \leqslant M \times I_q(j) \\ & w(j) \geqslant u - M \times [1 - I_q(j)] \\ & u > 0, \quad I_q(j) \in \{0, 1\}, \quad w(j) \geqslant 0 \end{aligned} \tag{A.4}$$

Appendix A.3. Additional MILP constraints for cluster definitions

As our goal is to select information elements through cluster parameters that define spatial, temporal, and keyword coherence, we add three constraints to the above optimization to define each of these cluster criteria:

1. **Time Selection Constraint:** a two-element tuple (t_{start}, t_{end}) indicating respectively the start and the end of the time window.

$$I_{time}(j) = \begin{cases} 1, & \text{if } (t_{start} \leqslant t(j)) \wedge (t(j) \leqslant t_{end}) \\ 0, & \text{otherwise} \end{cases} \tag{A.5}$$

2. **Spatial Selection Constraint:** a four-element tuple (x_{min}, y_{min}, x_{max}, y_{max}) to create a bounding box selection in visualization interface.

$$I_{pos}(j) = \begin{cases} 1, & \text{if } (x_{min} \leqslant x(j)) \wedge (x(j) \leqslant x_{max}) \wedge \\ & (y_{min} \leqslant y(j)) \wedge (y(j) \leqslant y_{max}) \\ 0, & \text{otherwise} \end{cases} \tag{A.6}$$

[9] `https://optimization.mccormick.northwestern.edu/index.php/Mixed-integer_linear_fractional_programming_(MILFP)`

3. **Keyword Selection Constraint:** a boolean vector of terms t_k^* with size m - the size of the dictionary of the global collection.

$$I_{term}(j) = \bigwedge_{t_k^* \in j} t_k^* \qquad \text{for k = 1, 2, } \cdots \text{, m} \tag{A.7}$$

   All terms with $I_{term} = 0$ are included in the negation query.

4. **Global Selection Constraint:** for information element j to be selected globally, it must be simultaneously selected by the three selection parameters.

$$I_q(j) = I_{time}(j) \wedge I_{pos}(j) \wedge I_{term}(j) \tag{A.8}$$

We refer to the above MILP formulation in (A.4) with all selection constraints (A.5)–(A.8) as the **Optimal** relevance-driven cluster denoted **RadiCAL-MILP**.