Summarizing Sets of Related ML-Driven Recommendations for Improving File Management in Cloud Storage

Will Brackenbury
wbrackenbury@uchicago.edu
University of Chicago
USA

Kyle Chard
chard@uchicago.edu
University of Chicago
USA

Aaron J. Elmore
aelmore@uchicago.edu
University of Chicago
USA

Blase Ur
blase@uchicago.edu
University of Chicago
USA

ABSTRACT

Personal cloud storage systems increasingly offer recommendations to help users retrieve or manage files of interest. For example, Google Drive's Quick Access predicts and surfaces files likely to be accessed. However, when multiple, related recommendations are made, interfaces typically present recommended files and any accompanying explanations individually, burdening users. To improve the usability of ML-driven personal information management systems, we propose a new method for summarizing related file-management recommendations. We generate succinct summaries of groups of related files being recommended. Summaries reference the files' shared characteristics. Through a within-subjects online study in which participants received recommendations for groups of files in their own Google Drive, we compare our summaries to baselines like visualizing a decision tree model or simply listing the files in a group. Compared to the baselines, participants expressed greater understanding and confidence in accepting recommendations when shown our novel recommendation summaries.

CCS CONCEPTS

• **Information systems** → **Data management systems**.

KEYWORDS

cloud storage, recommendations, Google Drive, personal information management

ACM Reference Format:
Will Brackenbury, Kyle Chard, Aaron J. Elmore, and Blase Ur. 2022. Summarizing Sets of Related ML-Driven Recommendations for Improving File Management in Cloud Storage. In *The 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22), October 29-November 2, 2022, Bend, OR, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3526113.3545704

UIST '22, October 29-November 2, 2022, Bend, OR, USA

ACM ISBN 978-1-4503-9320-1/22/10…$15.00
https://doi.org/10.1145/3526113.3545704

1 INTRODUCTION

Managing personal information in cloud storage (e.g., Google Drive) can be challenging [9, 12, 27, 46, 60]. In response, widely deployed tools like Google Drive's Quick Access [18, 84] and research prototypes [13, 47, 97] use machine learning (*ML*) to recommend files that a user may wish to view, delete, or move. To date, such recommendations have been based on characteristics like temporal patterns in the user's historical interactions with that file [40, 97], the other users with whom the file is shared [47], and the user deleting or moving other files that appear similar [13]. To help the user understand the recommendation, these tools typically provide a short explanation, such as "… because you *edited resume2022.docx* on *2022-04-07*" [41, 48, 66, 87, 97].

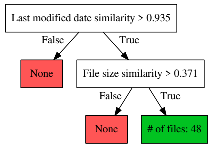
Even though a user's cloud storage repository typically contains many related files [12], resulting in the tools' ML models concurrently producing highly related recommendations for highly similar files, current tools and prototypes generally make recommendations individually for a single file at a time. Failing to aggregate groups of related recommendations increases the burden on users.

In this paper, we thus investigate whether related ML-driven recommendations for managing similar files in cloud storage can be aggregated effectively. This goal produces challenges related to both the underlying algorithm and the user experience. First, recommendations must be clustered into groups that a user would perceive as actually related, and the algorithm for doing so must be efficient. Second, the system must produce and display a succinct summary of the recommended files that enables the user to determine accurately which files are being recommended, a task we, and prior work [66], term **verification**.

Intuitively, files with similar **attributes** (e.g., filenames, file extensions, contents, location) that are being recommended for similar reasons are likely candidates for aggregation into a single recommendation that applies to multiple files at once. To this end, we first propose an algorithm (Section 3) for summarizing related files based on these shared file attributes. The algorithm takes as input a **group of recommendations**, or multiple files with similar attributes for which the same action — viewing the files, deleting them, or moving them to a specific location — is recommended for the same reason (e.g., a particular file was deleted). As output, the algorithm produces a set of predicates that apply to all files in a modified set of recommendations. While a naive approach would have computational complexity exponential in the space of file

Filename
Approaches for Assessing ...
Appendix 1 version 2.pdf
Appendix 2.docx
Approaches for Assessing ...
w21catalog HIGHLIGHTED.pd...
Approaches for Assessing ...

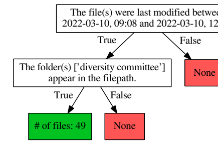
(a) Table listing all files in a group.



(b) *Decision Tree* summary

The system is recommending every file that matches the following criteria:
The file(s) were **last modified** between 2022-03-10, 09:08 and 2022-03-10, 12:33
AND
The folder(s) ['Diversity Committee'] **appear in the filepath**

(c) *Rules-Text* summary



(d) *Rules-Tree* summary

Figure 1: To communicate to users which files are contained in a group of recommendations, the most naive approach was to simply list the files (far left). Our summaries augmented this list with either a decision tree (center-left) as a baseline or the rule-based summaries we propose in either text-based (center-right) or tree-based (far right) presentations.

attributes, we develop a greedy approximation algorithm that takes roughly one second on commodity hardware.

The second challenge is to create a representation that helps the user understand which files are included in the group. The most basic approach would be to simply list the files and their most relevant metadata in a table in the user interface. However, this approach is unlikely to scale meaningfully to groups of recommendations that contain many files, and it also does not give any indication about what types of files are excluded from the group. As a result, we develop user-facing **summaries** that leverage our algorithm's output: the shared attributes of all files in the group (e.g., all documents whose filenames start with 'group-work' and that were modified within a particular date range). We design a text-based summary, termed ***Rules-Text*** and shown in Figure 1(c), and a visual tree-based summary, termed ***Rules-Tree*** and shown in Figure 1(d).

To evaluate our summaries, we conduct a within-subjects online user study (Sections 4–5). We show participants groups of recommendations about their own Google Drive repositories and solicit their perceptions of the associated summaries. We compare the aforementioned *Rules-Text* and *Rules-Tree* summaries we developed with two baselines: simply showing a table listing the files in the group, termed *List of Files* and shown in Figure 1(a), and a decision tree, termed *Decision Tree* and shown in Figure 1(b). We chose the latter since decision trees are often considered among the most interpretable ML classifiers [57].

We find that participants perceive our rule-based summaries as less confusing, more helpful, and more verifiable than the two baselines regardless of the number of recommendations in the group. In particular, compared to *List of Files* summaries, we find that *Rules-Text* summaries are 2.7× as likely to have a higher participant rating of helpfulness or verifiability. Further, compared to *List of Files* summaries, *Rules-Text* summaries are 2.0× as likely to have a higher participant rating of confidence in accepting recommendations without examining the individual files. Contrary to our expectation that participants would prefer visual displays, participants rate our text-based summaries slightly better than our tree-based summaries. We conclude (Section 6) by discussing implications for designing user interfaces that group and summarize recommendations for managing cloud storage repositories.

2 RELATED WORK

In this section, we provide an overview of prior work in set summarization, AI explanations, and personal information management that informs our research.

2.1 Set Summarization

Researchers have summarized sets of items in numerous ways. Some techniques summarize with a representative subset of the items, such as centroid approaches [54], top-k [15], regret minimization [45], KL-divergence [98], maximum entropy [91], or Bayesian Information Criterion [58]. We avoid such techniques due to their low verifiability. Other techniques extract feature information to generate a plaintext summary, as in text summarization [99] and image captioning [39]. These summaries, however, are also unlikely to be verifiable and are generated via a training set of existing summaries, which are not available. Alternatively, researchers have used application-specific visualizations to represent the item space [19, 42]. These visualizations, however, require global consistency across different summaries, while we do not. This allows for more succinct summaries that are more efficient to synthesize. Similar work that has visually represented local summaries has not been generalized to the setting of multiple recommendations [73]. We borrow parts of these prior works by incorporating a hover interaction into our visual explanations (*Decision Tree* and *Rules-Tree*) that shows what files are covered by a predicate of the summary.

More closely related to our techniques are summaries using tables of attributes [30, 94]. Our rules-based summaries extend these by also generating predicates over set-typed data. Similar to summary tables, associative rules for frequent itemsets [2, 11] and their related techniques for classification [26, 53] seek to generate and describe relationships over related items. These techniques, like the visual explanations described above, require global consistency.

It is less common for set summarization to have been applied to the domain of recommender systems. The closest analogues are in conversational recommender systems, where some researchers summarize how the set of unexplored items differs from the set of explored items [16]. Researchers have augmented this to describe categories of unexplored items based on extracted review sentiment [17]. Other work, while it does not investigate summaries, has focused on related sets of recommendations, which it dubs "slate" recommendations [63, 82]. Our work can be interpreted as seeking methods to summarize these slates.

2.2 AI Explanations

Our recommendation summaries generalize explanations in AI systems [1, 25, 36, 48, 55, 66, 88]. Explanations have been shown to improve users' understanding [81] and trust [29] in a system and help teach users when a system can be relied upon to make accurate judgments [51, 74]. Many explanation types are based on "interpretable" models, such as sparse linear classifiers [73], rule sets [92, 93], trees [57], or programs [80]. Our proposed summaries bear a strong resemblance to rule set explanations. We adapt these to the setting of file recommendation and make two improvements: we do not require pre-mining predicates, and we present our explanations in plaintext [14, 64]. The predicates in our summaries also resemble short programs (e.g., a Python function) [80]. Work in this area informs our technique (described in Section 3) of modifying the group of recommended items post-hoc [37, 69]. Given that our target users are non-technical, though, we avoid programming syntax. We compare directly against decision-tree-based explanations [57] in our online study as these are a proxy for many "interpretable" models. As noted by Lipton [55], the interpretability of such models may be overstated–we discuss this in Section 3. While other works have augmented interpretable models in ways that compare closely to our own work [35], we differ from these in our generation of set-based predicates (also described in Section 3).

Researchers have also studied explanations in recommender systems [87, 100]. Beyond the aforementioned property of verifiability, known also as "scrutability" [24, 86] or "simulatibility" [55], prior work has proposed other evaluation metrics. Some are based on users' perceptions of explanations [81] or the improvement in user satisfaction [83]. Others are task-based, such as an explanation's ability to justify a recommendation [90], to help users hone in on their preferences [10], to enhance their understanding of available items [32], to persuade them [23, 38], or to increase their speed [61].

2.3 Personal Information Management

The lessons of personal information management in other settings translate to the cloud. Researchers have studied how users acquire information ("foraging") [7, 49, 70], store it [43], and subsequently "curate" it [67, 95]. Users typically do this to re-find the information more easily [3, 4, 22, 85]. Re-finding, however, can be difficult: Whittaker et al. found low rates of success for research participants attempting to re-find family photos [96], and Elswiler et al. described how participants often searched first through incorrect folders or submitted fruitless search queries before retrieving emails [31].

Supporting re-finding has therefore been a key goal of file management tools. File management recommendation systems such as those found in Google Drive [18, 84] or Microsoft OneDrive [97] are closely related to this work. The work by Xu et al. [97] on Microsoft OneDrive evaluates explanations of file retrieval recommendations. However, they neither investigate groups of related recommendations nor recommendations of behaviors beyond retrieval. Besides recommendations, tools can provide navigational assistance by offering shortcuts of paths to files [5, 6, 56], or by highlighting icons of items likely to be clicked when retrieving files [33, 52, 78, 89]. Researchers have also supplemented interfaces to better support curation, which subsequently improves retrieval. Offering the ability to attach tags to files [8, 21] and improving search and indexing capabilities [20, 28, 40, 59, 72, 75, 76] are key strategies. While these tools are effective at aiding retrieval, they do not improve a repository's underlying disorganization. Tools like those from Bergman et al. [9] and Segal and Kephart [77] buck this trend by providing tools that suggest folders to save either cloud files or emails, respectively. Brackenbury et al. [13] similarly offer recommendations that fully support file movement and deletion actions, as well as retrieval. None of the efforts to support richer file management support, however, study the effect of summaries.

3 SUMMARIZATION ALGORITHM

Here, we describe the motivation for generating summaries, our target format for summaries, and the associated algorithm we created for clustering and summarizing recommendations.

3.1 Motivation and Existing Summaries

Summarizing a group of recommendations is necessary to communicate to the user which files are included in the group, and which are excluded. While summaries are useful for file retrieval (viewing a file), they are even more important for destructive and permanent actions like deleting or moving files. This observation is notable since recent research has increasingly focused on tools to help users delete and move files to improve personal information management [9, 13, 27, 47]. Furthermore, even if multiple recommendations for file retrieval were summarized, the user would likely still view those files individually and sequentially, in contrast to bulk file deletion or bulk file movement.

If multiple recommendations are grouped and summarized in a way that the user trusts to convey which files are included, the user can accept them together, improving efficiency and increasing the user's confidence that related files have not been inadvertently excluded from the recommendation. Our summaries thus aim to empower users to quickly determine which files are covered by a summary, a task we call *verification*, with the associated property "verifiability" [66], "scrutability" [86], or "simulatibility" [55].

We evaluate four summary types: *List of Files*, *Decision Tree*, *Rules-Text*, and *Rules-Tree*. The former two are intended as baselines, whereas the latter two are novel contributions of this work. For the first baseline, summaries for file recommendation in current systems generally appear in the following form: "You performed {action} to {file name} in {time period}" [41]. We mirrored this phrasing in our *List of Files* baseline, and we also accompanied it (and all other summaries) with a table listing the files in the group, as shown in Figure 1(a). We expected these summaries to fall short when recommending that the same action be applied to multiple files. The user might wonder how the files listed relate to each other, or whether files with similar attributes were mistakenly excluded.

Our second baseline is based on an observation from efforts in interpretable ML. Decision tree classifiers are typically considered among the most intelligible types of ML models [57]. In particular, our *Decision Tree* baseline displays a visual tree-based representation of a decision tree classifier that is used to select files for the group of recommendations based on their similarity to a file spawning the recommendations (e.g., deleting NorthernLights_98.jpg might spawn recommendations to delete other, related files). We did not use a purely text-based *Decision Tree* condition (i.e., similar to *Rules-Text*) as such conditions performed poorly in initial pilot testing due to confusion resulting from their branching structure. As shown in Figure 1(b), the visualization of the decision tree references the kinds of information used by the classifier (e.g., a normalized quantification of the similarity of file names). Despite the inherent interpretability of a decision tree, we expected that

Table 1: The structure of our proposed summaries.

Summaries	P	::=	$(r \mid s) \wedge \ldots \wedge (r \mid s)$
Range Predicate	r	::=	$n_1 \leq x \leq n_2$
Set Predicate	s	::=	$(c_1 \in x) \wedge \ldots \wedge (c_n \in x) \mid s \vee s$

the model parameters would prove somewhat unintelligible to non-experts. This is because understanding whether a file would be recommended or not could require a complicated calculation for a non-technical user due to the featurization needed to improve classifier performance [55]. Our *Decision Tree* baseline is a direct application of techniques from the relevant literature [73]. While one could likely improve performance of this baseline by hand-crafting features that are less subject to the downside of low verifiability, this does not allow the technique to generalize to any black-box model, in contrast to our *Rules-Text* and *Rules-Tree* methods.

3.2 Structure of Rule-based Summaries

Table 1 details the format of the rule-based summaries we developed: *Rules-Text* and *Rules-Tree*. These summaries consist of the intersection of multiple predicates on the attributes of the files in the group (Table 2) presented in ways we designed to be interpretable to non-technical users. Intuitively, these predicates represent attributes of the files included in a group of recommendations. These predicates take two forms depending on the data type of the attribute. For numeric attributes, such as the file size or last modified date, the predicate covers a range of values (e.g., "files between 3 and 5 megabytes"). For set-based attributes (all others, such as the set of objects recognized in an image), the predicate evaluates to true if, for at least one of the subsets of items ("tokens") in the predicate, the file's relevant feature set contains all of the given items. For example, if a predicate on filename tokens takes the conjunction of the sets ["course", "2019"] OR ["course", "2020"], then any file with filename tokens containing either subset will be covered by the predicate. Tokens are generated for each text attribute by breaking at common text delimiters, and for *Recognized Objects* using a standard ResNet object detector. We take the union over tokens for all files as our potential tokens to use in summaries. To limit the computational cost and ensure simplicity of summaries, we allow no more than a single "OR" conjunction for a particular feature predicate. We also do not allow "OR" clauses between predicates / different features (e.g., "The folder(s) ['work'] appear in the file path OR the filename(s) start with 'budget_' "). As these design choices were based on an ad-hoc examination of pilot testing data, future work could relax these requirements. Given that the notion of similarity has been shown to strongly inform desires about richer file management actions [12] and because the displayed predicates appear readily verifiable, they seem to address the expected drawbacks of the baselines above.

Because we were also interested in how the summary was presented to users, we developed and tested two visual presentations for rule-based summaries. The *Rules-Text* summary shows a plain-text representation, as in Table 1, with minor embellishments (e.g., bolding) for readability. The *Rules-Tree* summary inserts predicates into the same tree structure used in our *Decision Tree* baseline.

3.3 Synthesis Algorithm

Synthesizing summaries in the form of Table 1 over multiple recommendations faces several challenges. First, the synthesized summary is highly unlikely to be able to exactly match the group of recommendations output by the original recommender system—approximating the output of a black-box model trades recommendation efficacy for interpretability. This is only a minor concern in prior work, as researchers either tune the neighborhood around a single example to be summarized such that summaries are rarely untruthful [73] or assume a particular model form for the recommender system [62, 79, 101]. We instead modify the set of recommendations included in a group to exactly match those covered by the summary. We hypothesize that this is a potentially beneficial form of regularization on the recommender system output. This is motivated by techniques from program synthesis [37, 69], but is, to our knowledge, novel in this space. It avoids the issue of summaries that do not match the recommended files, but it may generate sets of recommendations that are less desirable than the original set. We discuss this further in Section 5, though leave a deeper examination to future work. It is still desirable to match the original set of recommendations in a group as closely as possible. To do this, we select among summary candidates using the F_β score, calculated as a weighted harmonic mean of precision and recall, with weights set in pilot testing. Precision and recall are calculated by taking the files covered by the summary (final values of FR and FO in Algorithm 1) and comparing against the ground truth labels (the original recommendations). The set of files covered by a set of predicates is identified via pre-built sorted range or reverse-index data structures that enable efficient lookup. Second, finding a globally optimal candidate for set-based predicates may require enumerating an exponential number of candidates in the worst case. To address this, our synthesis algorithm greedily adds tokens to the potential set predicate. This takes time $O(nk)$, where n is the number of possible tokens to explain over, and k is the number of tokens in the optimal predicate. We find that k is usually small (< 5) in practice. In addition, we limit the number of tokens examined per file to 1,000 for our experiments. Future work may examine the practicality of this limit. Third, to integrate seamlessly with the underlying recommender system, summaries must be generated in close to real time. Thus, we compute an approximation by greedily selecting the best predicate to add to the current set.

With these challenges in mind, we synthesize summaries using Algorithm 1, which takes Algorithm 2 as a subroutine. Informally, Algorithm 1 looks at each attribute, and uses a subroutine to identify the best predicate for that attribute given the current set of items covered by the summary. Whichever one yields the most improvement in the F_β score is added to the summary. The algorithm halts when adding a predicate on another attribute would negatively impact the score. The best candidate for set-based attributes is approximated with Algorithm 2, while the best candidate for attributes that take range-based predicates is found by enumerating all choices. Building the data structures and enumerating solution candidates are viable in practice because the universe of constants for range predicates and tokens to be added to set predicates is restricted to values drawn from the original group of recommendations. Intuitively, choosing a value in a range predicate that was

Table 2: File attributes used in summaries, their predicate types, and sample text representations. Attributes were generated by taking unfeaturized versions of features from the classifier in [13] and iteratively pilot testing extractible attributes.

Attribute	Predicate Type	Example
Filename Prefix	Set	The filename(s) start with 'bronze-age'
Filename Tokens	Set	The filename(s) contain sub-part(s) ['group', 'work']
File Extension	Set	The file(s) have the extension 'png'
File Path	Set	The folder(s) ['useful'] appear in the filepath
Shared Users	Set	The file(s) are shared with ['example@gmail.com']
Recognized Objects	Set	The system thought it saw the object(s) ['website', 'letter'] in the image(s)
File Text Tokens	Set	The file data contains the word(s) ['earnings', 'call']
File Size	Range	The file(s) have size from 2.0 Kb to 1.0 Mb
Last Modified Date	Range	The file(s) were last modified between 4/7/2019 14:40 and 4/8/2019 14:45

not drawn from a recommended item cannot improve more than one drawn from a recommended item and can only negatively impact precision. A similar principle holds for tokens in sets that do not apply to any files in the group. While we find that the given synthesis algorithms are efficient in practice, we do not explore the optimality gap due to approximation, nor do we explore the potential for more efficient implementations.

Algorithm 1 Full Approximation algorithm

```
procedure FullApprox(Files in Recommendations, Other Files)
    FR ← Files in Recommendations, FO ← Other files
    summary ← []
    while summary has not used all attributes do
        P ← [], S ← []
        for each remaining unused attribute att do
            if att is set-based then
                predicate, score ← SetGreedy(att, FR, FO)
            else
                predicate, score ← max(valid predicates)
            Add predicate to P and score to S
        if max(S) ≥ 0 then
            Add P[argmax(S)] to summary
            fz ← files covered by P[argmax(S)]
            FR ← FR ∩ fz, FO ← FO ∩ fz
        else break
    return summary
```

Algorithm 2 SetGreedy

```
procedure SetGreedy(attribute, FR, FO)
    predicate ← []
    while predicate has not used all set elements do
        scores ← [], fz ← files covered by predicate
        for each token t in possible tokens for attribute do
            fz′ ← files covered by predicate ∪ el
            oldScore ← FBeta(FR ∩ fz, FO ∩ fz)
            newScore ← FBeta(FR ∩ fz′, FO ∩ fz′)
            Add newScore − oldScore to scores
        if max(scores) > 0 then
            Add tokens[argmax(scores)] to predicate
        else  break
    return predicate
```

4 METHODOLOGY

To study the effects of summary type, we conducted a two-part, within-subjects online user study. In Part 1, we scanned participants' Google Drive accounts, pre-computed groups of recommendations, and generated summaries of each of the four summary types for every group. We used stratified sampling to select up to 14 group / summary pairs that was presented in Part 2, asking participants to evaluate characteristics like their helpfulness and verifiability.

4.1 Part 1

We recruited crowdworkers from the USA and UK through Prolific [71]. We required that participants had completed 10+ submissions with a 95%+ approval rating and had Google Drive accounts that were 3+ months old and contained 100+ files. Once we recruited participants and they had consented to the research, they granted our web application access through OAuth 2 to scan their Google Drive files' data and metadata. Participants were then directed to a survey on their demographics and usage of cloud storage. Part 1 took approximately 15 minutes. Compensation was $5.00.

We pre-computed file recommendations using Brackenbury et al.'s method [13]. Specifically, we calculated relevant data / metadata similarity features for a logistic regression classifier on pairs of files. We limited computation to all pairs of at most 1,000 files chosen uniformly at random. We generated groups of recommendations by iterating over all files, sequentially designating each as the "base file." All files classified as similar to the base file were recommended as a group. To limit overlap, we did not generate a group for the base file if that file appeared in a previous group. Intuitively, this mirrors a situation in which a user of the tool from Brackenbury et al. [13] performs an action on a file, and a large number of individual recommendations are generated and then aggregated into a group.

For each group, we then generated a summary of each type identified in Section 3 (*List of Files*, *Decision Tree*, *Rules-Text*, *Rules-Tree*). We excluded the base file from this summary as it was used to generate the "scenarios" described below. As described in Section 3, we modified the set of files in a group to exactly match those covered by the summary. The *List of Files* summaries require no generation, the *Rules-Text* and *Rules-Tree* summaries were generated with Algorithm 1 and the *Decision Tree* summaries were generated by training decision trees (Gini impurity, max depth of 2 set in pilot testing) that took the original group of recommendations as positive labels, and files not recommended as negative labels.

Once summaries were generated, we used stratified sampling to choose group / summary pairs to present in Part 2. We selected up to 14 groups as follows:

- 4 groups, based on summary complexity (2 "complex", 2 "simple")
- 4 groups, based on "discriminativeness" (2 "discriminative", 2 "non-discriminative")
- 6 groups, based on size (2 "small", ≤ 25th percentile of group size for participant, 2 "medium", 25th–75th percentile, and 2 "large", > 75th percentile)

We labeled *Rules-Text* or *Rules-Tree* summaries as complex if they required at least one 'AND' or 'OR' keyword, and *Decision Tree* summaries as complex if the resultant tree had depth > 1. *List of Files* summaries were not complex. We identified groups as discriminative based on what percentage of the files in a folder were recommended, among folders that contained recommended files. Intuitively, recommendations that suggest performing an action on all files in a folder (recommendations that are not "discriminative" of files in a folder) are less helpful for users, given that such files can easily be identified by the user themselves. In contrast, selecting a specific subset of files from a folder may require more effort from a user, and such recommendations are therefore more helpful. If there were fewer group / summary pairs that met the complex and discriminative criteria than desired, additional summaries were sampled from the small, medium, and large groupings.

4.2 Part 2

We invited back eligible participants after we had finished the processing of Part 1. We presented them with 14 hypothetical "scenarios" (the **Scenario**), based on a group / summary pair, each of which read, "Suppose that you shared, moved, or deleted {base file}". We presented the group of recommendations (**Recommended Files**) in a table with relevant metadata that linked to the file data in Google Drive (Figure 1(a)), along with the summary (the **Explanation**). While we refer to summaries as the **Explanation** to enhance participant understandability, we note this terminology is potentially misleading: our summaries are generated post-hoc without examining the internals of the black box classifier. For *List of Files* summaries, we presented only the text, "Because you shared, moved, or deleted {base file} ({file path of base file})". Other summary types were displayed as in Figure 1. The visual summary types, *Decision Tree* and *Rules-Tree*, also had a hover interaction on leaf nodes that displayed the names of the files allocated to that node. We then asked participants a set of 8 questions (shown in Table 3) about the scenario, group, and summary. Part 2 took approximately 1 hour, and compensation was $15.00.

4.3 Limitations

Our study required that participants accept permissions allowing our web application to view and download their file data—although our institution's IRB approved the study, privacy-conscious participants may have been unwilling to participate. In addition, our study presents hypothetical scenarios. While this allows us to directly study groups of file recommendations, participants' survey responses may be biased either towards accepting recommendations, because there was no cost to agreeing, or against accepting them, because of the uncertainty introduced by lack of context. Further, although it was necessary from a computational standpoint, limiting our all-pairs similarity to 1,000 files may bias our results. The absence of recommendations that would have been included, had the files been sampled for similarity, may negatively bias participants' survey responses for summary types based on pre-computed similarity (*List of Files*, *Decision Tree*). Our study was also conducted on crowdworkers. Prior work has shown that crowdworkers are not representative of any broader population, and that many skew younger and more technically-savvy [68].

5 RESULTS

We describe our participants and their survey responses, then build a set of regression models to identify the effect of summary type on qualities such as understandability, helpfulness, and verifiability.

5.1 Participants

44 participants completed both parts of our within-subjects user study. 29 (67.4%) participants were female, 11 (25.6%) were male, and 3 (7.0%) were non-binary. Most participants were 25–34 years old (16, 36.4%), with a similar number (15, 34.1%) 18–24 years old, and the remaining 35–64 years old. Most (35, 79.5%) had no computer science background. Participants interacted with their Google Drive account in various ways. Participants used Google Drive through the website (37) or the mobile app (30) nearly equally, though a few synced folders directly from their local storage (12). Most participants interacted with their account weekly (17, 40.4%), though monthly (13, 31.0%) and daily (11, 26.2%) usage was also common. Participants generally disagreed that their accounts were well-organized (15, 34.1% "Disagree", and 13, 29.5%, "Strongly disagree"). Participants also generally agreed that their files were "uncategorized" (15, 34.1%, "Agree" and 13, 29.5%, "Strongly agree").

The distribution of participants' cloud storage files was similar to analogous populations from prior work [12, 13]. We processed 97,546 files from participants. The median participant had 1,310.5 files in their account, and the mean participant had 2,217 files, with a standard deviation of 3,622.5. The smallest account had 117 files, and the largest, 16,137 files. Most files were images (43,889), with a large number of media (14,791) and text files (14,199) across participants. Most images were "jpg" files (35,085), most media files were "mp3" (4,164) or "heic" files (4,049), and most text files were "pdf" files (9,790). There was also a long tail of 23,172 files with uncategorized extensions. These included "no extension" (5,131), Autodesk files ("flc", 1,893) and paintbrush bitmap files ("pcx", 651).

5.2 Survey Responses

Participants saw 563 scenarios. Summary types appeared in roughly equal numbers of scenarios: 131 (23.3%) *List of Files* scenarios, 153 (27.2%) *Decision Tree* scenarios, 132 (23.4%) *Rules-Text* scenarios, and 147 (26.1%) *Rules-Tree* scenarios. The sampling reasons were roughly evenly distributed as well. The most common choice was summaries over small file groups (90, 16.0%), and the least common choice was non-discriminative summaries (64, 11.4%). The size of the recommendation groups followed roughly a power-law distribution: the mean sampled group contained 40.2 recommendations, while the median sampled group contained 7 recommendations.

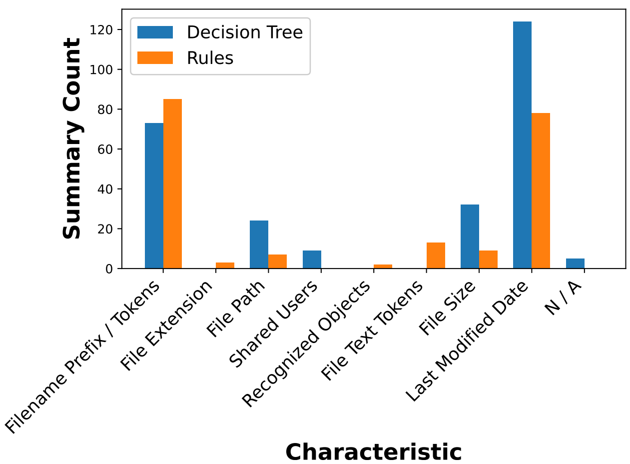


Figure 2: Number of times each attribute appeared in a summary for *Decision Tree*, or a *Rules-Text*/ *Rules-Tree*. "N/A" represents *Decision Tree* features not available for rules.

The largest group sampled was 1,179 recommendations. On average, groups identified by *Rules-Text* and *Rules-Tree* summaries were larger: groups had median size 9 for both, compared against median sizes of 6 for *List of Files* and *Decision Tree* summaries, respectively. However, per the discussed limitation in Section 4.3, this is likely to be biased. Differences between summary types carried over to the scores: *Decision Tree* summaries had an average F_β score of 93.0, while *Rules-Text* and *Rules-Tree* summaries had a score of 68.9. This is a notable difference, but there are several considerations. First, again due to the limitation in Section 4.3, scores for *Decision Tree* summaries are biased upward, as they are fitting a smaller set of files. Second, scores only indicate a summary's ability to match the original classifier recommendations. This is independent of participants' perceptions of the recommendations and summaries, which is the focus of our analysis.

The distributions of file attributes (Table 2) chosen for summarization were similar across summary types, as seen in Figure 2. We display only *Decision Tree* and *Rules-Text* summaries, as *List of Files* summaries do not use attributes, and *Rules-Tree* summaries are syntactically equivalent to *Rules-Text* summaries. By far, summaries most commonly used filenames and last modified dates. File path and file size attributes occasionally appeared, and summaries rarely used the remaining attributes. Some attributes were not present for a particular summary type, due either to non-extant features in the original classifier or impracticability of generating predicates for some classifier features. Such attributes were rarely used.

We display the proportion of Likert-scale responses for each question from Table 3 in Figure 3. We note the difference between two types of questions: "Group-Based" questions (Q1 & Q7) that could be answered without reference to a summary, and "Summary-Based" questions that asked about the summary specifically. We use "Summary-Based" responses to evaluate our core research questions. We use "Group-Based" responses both to analyze our recommendations compared to prior work [12, 13], and to control for summary-independent aspects in our regressions (Table 4). Responses to "Group-Based" questions roughly matched expectation from prior work. The responses to Q1 (> 50.0% "Agree" or "Strongly agree" responses) suggest participants generally found recommendation groups to be related. This approximately resembles the incidence of similar files under stratified sampling in prior work [12]. Importantly, we note that the proportion of "Strongly agree" or "Agree" responses for *Rules-Text* and *Rules-Tree* summaries were roughly equal to other summaries. As Q1 is summary-independent, the responses can be considered a proxy for the effect of post-hoc modifying the original recommendation set. The similarity across summary type, therefore, suggests that this technique is not noticeably harmful, though further investigation is needed. In Q7, participants indicated they would accept the group of recommendations ("Strongly agree" + "Agree") for between 1/3 and 1/2 of scenarios across summary type. This is comparable with, though slightly higher than, acceptance rates of similar individual recommendations observed in practice [13]. Future field studies of summaries will be most helpful in determining how this compares with group recommendation in practice.

Participants generally found our summaries (*Rules-Text* in particular) more understandable, less confusing, more helpful, and more verifiable than *List of Files* or *Decision Tree* summaries. The responses to Q2 suggest that participants could describe each summary type ("Agree" and "Strongly agree" responses > 50% across summary types). Pilot testing suggested Q2 was a reasonable proxy for "understandability". *Rules-Text* and *Rules-Tree* summaries have a higher proportion of "Strongly agree" or "Agree" responses than *List of Files* or *Decision Tree* summaries for this question: participants answered "Strongly agree" or "Agree" in 86.4% of scenarios for *Rules-Text* summaries, and in 74.0% for *Rules-Tree* summaries. Q3 shows a similar response distribution, with flipped sentiment due to the nature of the question. We examine the significance of these responses when controlling for the relatedness of the files and participant-specific effects in Section 5.3. For Q4, participants seemed to find *Decision Tree* summaries less helpful, only responding "Strongly agree" or "Agree" in 28.8% of scenarios. This is surprising, given that *Decision Tree* summaries are widely used in literature, and the *List of Files* baseline is very simple. This potentially suggests that *Decision Tree* summaries present information that distracts users. Future work may wish to examine what aspects of *Decision Tree* summaries are unhelpful and in what situations. Participants indicated that *List of Files* summaries were helpful in 40.5% of scenarios, *Rules-Text* in 56.8% and *Rules-Tree* in 47.3%. The slightly lower rate of positive responses for *Rules-Tree* summaries compared to *Rules-Text* summaries, combined with the similarity in presentation between *Decision Tree* and *Rules-Tree* summaries offers some further evidence that participants considered the decision tree visualization style less helpful. The proportion of positive responses to Q5 for *Rules-Text* and *Rules-Tree* summaries compared to other summary types offers some evidence that such summary types were more verifiable. Participants responded "Strongly agree" or "Agree" for 68.2% of *Rules-Text* summaries, for 63.7% of *Rules-Tree* summaries, for 49.6% of *List of Files* summaries, and for 45.1% of *Decision Tree* summaries. Interestingly, despite the minimal information in *List of Files* summaries, participants appeared to believe they could still identify which files were covered by the summary.

Table 3: Questions shown to participants for each scenario in Part 2. We referred to groups of recommendations as "Recommended Files", the summary as the "Explanation", and the file action producing the recommendations as the "Scenario."

Q1:	The **Recommended Files** are related to each other
Q2:	I could accurately describe to someone else what the **Explanation** is saying
Q3:	The **Explanation** is confusing
Q4:	I'd find a style of of explanation similar to this **Explanation** helpful when files are recommended to me
Q5:	If I saw a table of all the files in my Google Drive, I could pick out which ones the **Explanation** covered
Q6:	Based on the **Explanation** given, I believe the system sees the **Recommended Files** as related for the same reasons I do
Q7:	I would perform the same action as in the **Scenario** on the **Recommended Files**
Q8:	After seeing the **Explanation**, I would feel more confident performing the same action as in the **Scenario** on the **Recommended Files** without examining every file individually

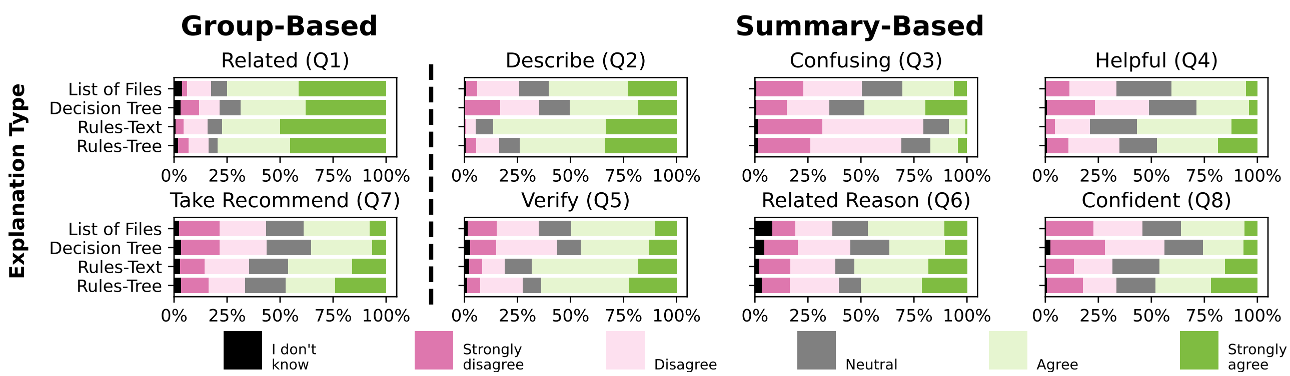


Figure 3: Proportion of Likert scale responses to each question, separated by summary type. "Group-Based" questions are those that are answered without reference to a summary, while "Summary-Based" questions referred explicitly to a summary.

Additionally, we find that participants indicated stronger confidence in a greater proportion of scenarios for *Rules-Text* or *Rules-Tree* summaries compared to others. Participants responded "Strongly agree" or "Agree" for 46.2% and 47.9% for *Rules-Text* and *Rules-Tree* summaries, respectively. In contrast, participants only responded such for 25.5% of scenarios with *Decision Tree* summaries and 35.9% of scenarios with *List of Files* summaries. In this case, despite the slightly lower support for *Rules-Tree* summaries indicated in questions such as the helpfulness of the style, *Rules-Tree* summaries were the type that participants found improved their confidence in the most scenarios. The answers to this question go hand-in-hand with those for Q5, as both are aimed at determining whether summaries helped participants make better / more informed decisions with groups of recommendations. We analyze whether this trend held when controlling for other factors below.

5.3 Regression Model

To disentangle correlated factors in the responses in Figure 3, we built a set of cumulative linked logit mixed effects regression models (Table 4). We chose this model format because Likert responses are ordinal and responses by the same participant are correlated.

We take the Likert rating of the "Summary-Based" questions as our response variables. For the models of Q2–Q6, the fixed effects are the presence of each summary type compared against the *List of Files* type, as well as the Likert response to Q1. This last factor is because participants will likely rate summaries more negatively if participants believe the files recommended are less related to each other. For **Confident (Q8)**, the Likert response from Q1 is changed for Q7, indicating whether a participant would accept the group of recommendations in the first place. If a participant is unlikely to accept a group of recommendations, the summary quality is irrelevant to their confidence in accepting the recommendations. We exclude from these models the size of the group of recommendations, and the reason a group was sampled, as these were not found to be statistically significant factors in any model where they were included. This potentially indicates that our results apply to recommendation groups of a range of sizes and with a variety of properties. Table 4 displays odds ratios, which are interpreted as the multiplicative increase in the odds that a higher Likert response is given for the dependent variable when a summary type is present or when the Likert response for a covariate is one point higher. For example, as seen in the first column of Table 4, a participant's response was roughly 2.7x more likely to be a higher Likert rating if a *Rules-Text* summary was provided as compared to a *List of Files*.

The only summary type that is statistically significant across all but one model is *Rules-Text*. Further, in each model, the effect direction is as expected: the odds ratio is > 1 (a multiplicative *increase*) for all questions where higher agreement indicates positive attributes, and < 1 for **Confusing (Q3)**, where lower confusion is preferred. The effect size is also notable: the presence of a *Rules-Text* summary has a 2.7× odds improvement for models Q2–Q5, and a 2.0× improvement for **Confident (Q8)**. The effect size, combined with the high statistical significance of the *Rules-Text* summary

Table 4: Cumulative link logit mixed effects regressions on the Likert responses for Summary-Based questions. Coefficients are odds ratios, interpreted as the multiplicative increase in the odds of a higher response. p-values were calculated based on the Satterthwaite method. Asterisks indicate level of statistical significance ($^{***} = p < 0.001$, $^{**} = p < 0.01$, $^{*} = p < 0.05$).

	Describe (Q2)	Confusing (Q3)	Helpful (Q4)	Verify (Q5)	Related Reason (Q6)	Confident (Q8)
Fixed Effects						
Related (Q1)	1.761^{***}	0.641^{***}	1.673^{***}	2.101^{***}	3.049^{***}	——
Take Recommend (Q7)	——	——	——	——	——	7.592^{***}
Decision Tree	0.553^{*}	2.729^{***}	0.571^{*}	0.945	0.842	0.774
Rules-Text	2.729^{***}	0.353^{***}	2.718^{***}	2.791^{***}	1.032	1.987^{**}
Rules-Tree	1.637^{*}	0.630^{*}	1.412	1.893^{**}	1.013	1.567
Random Effects						
Participant effect	1.169	1.110	1.618	1.318	1.586	1.370

variable, suggests that such summaries may carry a number of benefits: they may be more understandable, helpful, and confidence-inducing while being less confusing. While *Rules-Tree* summaries also showed some benefit compared to *List of Files* summaries, the effect size and statistical significance were lower. The *Decision Tree* variable in the regression models, when significant, was rated *lower* than baseline *List of Files* summaries: they were less often able to be described (Q2, 0.5x) or to be helpful (Q4, 0.5x) and were more often confusing (Q3, 2.7x). Given that *Rules-Text* and *Rules-Tree* summaries differed only in that *Rules-Tree* presented information like *Decision Tree* summaries did, this suggests that the *Decision Tree* format may require additional improvements to be competitive with other approaches along the same metrics. We leave the specifics of these needed improvements to future work. Interestingly, the sole model where no summary type had a statistically significant effect was **Related Reason (Q6)**. One interpretation is that, though summaries could be effective at helping participants verify inputs, they may have differed from the participants' mental model of the identified files. Future work may wish to examine this effect when summaries are incorporated into full tools. We additionally find that the participant-specific effect for a model was, on average, about a point to a point-and-a-half difference in Likert response. This suggests that even independent of the relatedness of recommendations or the summary type presented, participants still responded to scenarios very differently. This suggests that future work on sets of related recommendations may find significant benefit in personalization of recommendations [62].

6 DISCUSSION

We proposed and evaluated a new way of summarizing groups of file management recommendations in cloud storage. We also presented an efficient approximation algorithm to synthesize these summaries. We conducted a 44-participant, within-subjects online user study in which we compared our newly proposed summaries (*Rules-Text* and *Rules-Tree*) against baselines (*List of Files* and *Decision Tree*). Compared to our baselines, participants were more likely to rate *Rules-Text* summaries as more verifiable and more confidence-increasing when considering a groups of recommendations without examining individual recommendations.

Future interfaces supporting file management recommendations may take two main lessons from our work. First, summarizing groups of recommendations is feasible. Though summaries are not provided by current cloud storage systems, our techniques show they can be added without significant computational overhead. Further, participants' ability to understand such summaries was high across summary types. While summaries may be less beneficial for file-retrieval recommendations, they may be valuable for more complex file management actions. Summaries could potentially even be useful for multi-round recommendation [62, 65] by increasing user understanding of available items up-front, instead of gradually revealing this information through multi-round interaction.

The second lesson is that *Rules-Text* summaries can offer users the ability to verify files in recommendation groups. This potentially relates to participants' higher confidence when accepting recommendations from *Rules-Text* summaries: knowledge of a file collection combined with verifiability allows a user to compute what files are included in a recommendation group without examining directly. The verification and increased confidence are likely the most important properties for summaries, given the use case. We hypothesize that the key attributes of *Rules-Text* summaries that produced this verifiability were their plaintext representation, and the predicates with minimal featurization. The first of these is evidenced by the lower ratings of the *Rules-Tree* summaries compared to *Rules-Text* summaries, as well as the more-negative ratings for *Decision Tree* summaries than for *List of Files* summaries. However, future work should investigate several caveats. First, the strength of *Rules-Text* summaries may not translate to real deployments and other types of summaries might be preferable. For example, because our *Decision Tree* summaries used highly-featurized inputs from the black box classifier, classifiers with less featurization might find that *Decision Tree* summaries compare more favorably. Alternatively, interface-specific effects such as summary presentation might outweigh the effects found here [34]. Lastly, measurements of verifiability and confidence may prove to be uncorrelated with desired user behavior. The enhanced interaction allowed by rules-type summaries, though, is a strong benefit for usability. In work like SmallStar [50] and Wrangler [44], for example, users iteratively specify short programs within a given framework. Systems could provide similar interactions based on rules-like summaries, offering new modes of interaction in file recommendation settings.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grants No. CNS-1801663 and OAC-1835890.

REFERENCES

[1] Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE Access* 6 (2018).

[2] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. 1993. Mining association rules between sets of items in large databases. In *Proc. SIGMOD*.

[3] Tarfah Alrashed, Ahmed Hassan Awadallah, and Susan Dumais. 2018. The lifetime of email messages: A large-scale analysis of email revisitation. In *Proc. CHIIR*.

[4] Anne Aula, Natalie Jhaveri, and Mika Käki. 2005. Information search and re-access strategies of experienced web users. In *Proc. WWW*.

[5] Xinlong Bao and Thomas G. Dietterich. 2011. FolderPredictor: Reducing the cost of reaching the right folder. *ACM TIST* 2, 1 (2011).

[6] Xinlong Bao, Jonathan L. Herlocker, and Thomas G. Dietterich. 2006. Fewer clicks and less frustration: Reducing the cost of reaching the right folder. In *Proc. IUI*.

[7] Nicholas J. Belkin. 1980. Anomalous states of knowledge as a basis for information retrieval. *Canadian Journal of Information Science* 5, 1 (1980).

[8] Ofer Bergman, Noa Gradovitch, Judit Bar-Ilan, and Ruth Beyth-Marom. 2013. Tagging personal information: A contrast between attitudes and behavior. In *Proc. ASIS&T*.

[9] Ofer Bergman, Steve Whittaker, and Yaron Frishman. 2019. Let's get personal: The little nudge that improves document retrieval in the cloud. *J. Doc* (2019).

[10] Mustafa Bilgic and Raymond J. Mooney. 2005. Explaining recommendations: Satisfaction vs. promotion. In *IUI Beyond Personalization Workshop*, Vol. 5.

[11] Christian Borgelt. 2012. Frequent item set mining. *WIREs: Data Mining and Knowledge Discovery* 2, 6 (2012).

[12] Will Brackenbury, Galen Harrison, Kyle Chard, Aaron Elmore, and Blase Ur. 2021. Files of a feather flock together? Measuring and modeling how users perceive file similarity in cloud storage. In *Proc. SIGIR*.

[13] Will Brackenbury, Andrew McNutt, Kyle Chard, Aaron Elmore, and Blase Ur. 2021. KondoCloud: Improving information management in cloud storage via recommendations based on file similarity. In *Proc. UIST*.

[14] Shuo Chang, F. Maxwell Harper, and Loren Gilbert Terveen. 2016. Crowd-based personalized natural language explanations for recommendations. In *Proc. RecSys*.

[15] Surajit Chaudhuri and Luis Gravano. 1999. Evaluating top-k selection queries. In *Proc. VLDB*.

[16] Li Chen and Pearl Pu. 2007. Preference-based organization interfaces: Aiding user critiques in recommender systems. In *Proc. UMAP*.

[17] Li Chen and Feng Wang. 2017. Explaining recommendations based on feature sentiments in product reviews. In *Proc. IUI*.

[18] Suming Jeremiah Chen, Zhen Qin, Zachary Teal Wilson, Brian Lee Calaci, Michael Richard Rose, Ryan Lee Evans, Sean Robert Abraham, Don Metzler, Sandeep Tata, and Mike Colagrosso. 2020. Improving recommendation quality at Google Drive. In *Proc. KDD*.

[19] Yen-Liang Chen and Lucas Tzu-Hsuan Hung. 2009. Using decision trees to summarize associative classification rules. *Expert Systems with Applications* 36, 2 (2009).

[20] Paul-Alexandru Chirita, Stefania Costache, Wolfgang Nejdl, and Raluca Paiu. 2006. Beagle++: Semantically enhanced searching and ranking on the desktop. In *Proc. ESWC*.

[21] Andrea Civan, William Jones, Predrag Klasnja, and Harry Bruce. 2008. Better to organize personal information by folders or by tags?: The devil is in the details. In *Proc. ASIS&T*.

[22] Andy Cockburn and Bruce McKenzie. 2001. What do web users do? An empirical analysis of web use. *IJHCS* 54, 6 (2001).

[23] Henriette Cramer, Vanessa Evers, Satyan Ramlal, Maarten Van Someren, Lloyd Rutledge, Natalia Stash, Lora Aroyo, and Bob Wielinga. 2008. The effects of transparency on trust in and acceptance of a content-based art recommender. *UMUAI* 18, 5 (2008).

[24] Marek Czarkowski and Judy Kay. 2002. A scrutable adaptive hypertext. In *Proc. AH*.

[25] Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A survey of the state of explainable AI for natural language processing. *arXiv:2010.00711*.

[26] Guozhu Dong, Xiuzhen Zhang, Limsoon Wong, and Jinyan Li. 1999. CAEP: Classification by aggregating emerging patterns. In *Proc. DS*.

[27] Dropbox. 2022. https://help.dropbox.com/files-folders/sort-preview/multi-file-organize.

[28] Susan Dumais, Edward Cutrell, Jonathan J. Cadiz, Gavin Jancke, Raman Sarin, and Daniel C. Robbins. 2003. Stuff I've seen: A system for personal information retrieval and re-use. In *Proc. SIGIR*.

[29] Mary T. Dzindolet, Scott A. Peterson, Regina A. Pomranky, Linda G. Pierce, and Hall P. Beck. 2003. The role of trust in automation reliance. *IJHCS* 58, 6 (2003).

[30] Kareem El Gebaly, Parag Agrawal, Lukasz Golab, Flip Korn, and Divesh Srivastava. 2014. Interpretable and informative explanations of outcomes. In *Proc. VLDB*.

[31] David Elsweiler, Morgan Harvey, and Martin Hacker. 2011. Understanding re-finding behavior in naturalistic email interaction logs. In *Proc. SIGIR*.

[32] Alexander Felfernig and Bartosz Gula. 2006. An empirical study on consumer behavior in the interaction with knowledge-based recommender applications. In *Proc. CEC/EEE*.

[33] Stephen Fitchett, Andy Cockburn, and Carl Gutwin. 2014. Finder highlights: Field evaluation and design of an augmented file browser. In *Proc. CHI*.

[34] Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. 2014. How should I explain? A comparison of different explanation types for recommender systems. *IJHCS* 72, 4 (2014).

[35] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. 2018. Local rule-based explanations of black box decision systems. *arXiv:1805.10820*.

[36] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM CSUR* 51, 5 (2018).

[37] Shivam Handa and Martin C. Rinard. 2020. Inductive program synthesis over noisy data. In *Proc. ESEC/FSE*.

[38] Jonathan Lee Herlocker. 2000. *Understanding and improving automated collaborative filtering systems*. University of Minnesota.

[39] MD Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. 2019. A comprehensive survey of deep learning for image captioning. *ACM CSUR* 51, 6 (2019).

[40] Rolf Jagerman, Weize Kong, Rama Kumar Pasumarthi, Zhen Qin, Michael Bendersky, and Marc Najork. 2021. Improving cloud storage search with user activity. In *Proc. WSDM*.

[41] Farnaz Jahanbakhsh, Ahmed Hassan Awadallah, Susan T. Dumais, and Xuhai Xu. 2020. Effects of past interactions on user experience with recommended documents. In *Proc. CHIIR*.

[42] Manas Joglekar, Hector Garcia-Molina, and Aditya Parameswaran. 2017. Interactive data exploration with smart drill-down. *IEEE TKDE* 31, 1 (2017).

[43] William Jones, Susan Dumais, and Harry Bruce. 2002. Once found, what then? A study of "keeping" behaviors in the personal use of web information. In *Proc. ASIS&T*.

[44] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. 2011. Wrangler: Interactive visual specification of data transformation scripts. In *Proc. CHI*.

[45] Taylor Kessler Faulkner, Will Brackenbury, and Ashwin Lall. 2015. K-regret queries with nonlinear utilities. In *Proc. VLDB*.

[46] Mohammad Taha Khan, Maria Hyun, Chris Kanich, and Blase Ur. 2018. Forgotten but not gone: Identifying the need for longitudinal data management in cloud storage. In *Proc. CHI*.

[47] Mohammad Taha Khan, Christopher Tran, Shubham Singh, Dimitri Vasilkov, Chris Kanich, Blase Ur, and Elena Zheleva. 2021. Helping users automatically find and manage sensitive, expendable files in cloud storage. In *Proc. USENIX Security*.

[48] Been Kim, Rajiv Khanna, and Oluwasanmi O. Koyejo. 2016. Examples are not enough, learn to criticize! Criticism for interpretability. In *Proc. NeurIPS*.

[49] Carol C. Kuhlthau. 1991. Inside the search process: Information seeking from the user's perspective. *JASIST* 42, 5 (1991).

[50] David Kurlander, Allen Cypher, and Daniel Conrad Halbert. 1993. *Watch what I do: Programming by demonstration*. MIT press.

[51] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2019. Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications* 10, 1 (2019).

[52] Bongshin Lee and Benjamin B. Bederson. 2003. *Favorite folders: A configurable, scalable file browser*. Technical Report.

[53] Wenmin Li, Jiawei Han, and Jian Pei. 2001. CMAR: Accurate and efficient classification based on multiple class-association rules. In *Proc. ICDM*.

[54] Aristidis Likas, Nikos Vlassis, and Jakob J. Verbeek. 2003. The global k-means clustering algorithm. *Pattern Recognition* 36, 2 (2003).

[55] Zachary C. Lipton. 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue* 16, 3 (2018).

[56] Wanyu Liu, Olivier Rioul, Joanna Mcgrenere, Wendy E. Mackay, and Michel Beaudouin-Lafon. 2018. BIGFile: Bayesian information gain for fast file retrieval. In *Proc. CHI*.

[57] Yin Lou, Rich Caruana, and Johannes Gehrke. 2012. Intelligible models for classification and regression. In *Proc. KDD*.

[58] Michael Mampaey, Nikolaj Tatti, and Jilles Vreeken. 2011. Tell me what I need to know: Succinctly summarizing data with itemsets. In *Proc. KDD*.

[59] Hannes Marais and Krishna Bharat. 1997. Supporting cooperative and personal surfing with a desktop assistant. In *Proc. UIST*.

[60] Charlotte Massey, Thomas Lennig, and Steve Whittaker. 2014. Cloudy forecast: An exploration of the factors underlying shared repository use. In *Proc. CHI*.

[61] Kevin McCarthy, James Reilly, Lorraine McGinty, and Barry Smyth. 2004. Thinking positively-explanatory feedback for conversational recommender systems. In *Proc. ECCBR Explanation Workshop*.

[62] James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. 2018. Explore, exploit, and explain: Personalizing explainable recommendations with bandits. In *Proc. RecSys.*

[63] Rishabh Mehrotra, Mounia Lalmas, Doug Kenney, Thomas Lim-Meng, and Golli Hashemian. 2019. Jointly leveraging intent and interaction signals to predict user satisfaction with slate recommendations. In *Proc. WWW.*

[64] Stephen Muggleton and Luc De Raedt. 1994. Inductive logic programming: Theory and methods. *The Journal of Logic Programming* 19 (1994).

[65] Danupon Nanongkai, Ashwin Lall, Atish Das Sarma, and Kazuhisa Makino. 2012. Interactive regret minimization. In *Proc. SIGMOD.*

[66] Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. 2018. How do humans understand explanations from machine learning systems? An evaluation of the human-interpretability of explanation. *arXiv:1802.00682.*

[67] Kyong Eun Oh. 2012. What happens once you categorize files into folders?. In *Proc. ASIS&T.*

[68] Gabriele Paolacci, Jesse Chandler, and Panagiotis G. Ipeirotis. 2010. Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making* 5, 5 (2010).

[69] Hila Peleg and Nadia Polikarpova. 2020. Perfect is the enemy of good: Best-effort program synthesis. In *Proc. ECOOP.*

[70] Peter Pirolli. 2007. *Information foraging theory: Adaptive interaction with information.* Oxford University Press.

[71] Prolific. 2019. https://www.prolific.co/.

[72] Dennis Quan, David Huynh, and David R. Karger. 2003. Haystack: A platform for authoring end user semantic web applications. In *Proc. ISWC.*

[73] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?" Explaining the predictions of any classifier. In *Proc. KDD.*

[74] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. 2017. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv:1703.03717.*

[75] Leo Sauermann, Gunnar Aastrand Grimnes, Malte Kiesel, Christiaan Fluit, Heiko Maus, Dominik Heim, Danish Nadeem, Benjamin Horak, and Andreas Dengel. 2006. Semantic desktop 2.0: The gnowsis experience. In *Proc. ISWC.*

[76] Markus Schröder, Christian Jilek, and Andreas Dengel. 2019. Interactive concept mining on personal data. *arXiv:1903.05872.*

[77] Richard B. Segal and Jeffrey O. Kephart. 1999. MailCat: An intelligent assistant for organizing e-mail. In *Proc. AGENTS.*

[78] Procheta Sen, Debasis Ganguly, and Gareth J.F. Jones. 2021. I know what you need: Investigating document retrieval effectiveness with partial session contexts. *ACM TOIS* 40, 3 (2021).

[79] Amit Sharma and Dan Cosley. 2013. Do social explanations work? Studying and modeling the effects of social explanations in recommender systems. In *Proc. WWW.*

[80] Sameer Singh, Marco Tulio Ribeiro, and Carlos Guestrin. 2016. Programs as black-box explanations. *arXiv:1611.07579.*

[81] Rashmi Sinha and Kirsten Swearingen. 2002. The role of transparency in recommender systems. In *Proc. CHI EA.*

[82] Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miro Dudik, John Langford, Damien Jose, and Imed Zitouni. 2017. Off-policy evaluation for slate recommendation. In *Proc. NeurIPS.*

[83] Kumiko Tanaka-Ishii and Ian Frank. 2000. Multi-agent explanation strategies in real-time domains. In *Proc. ACL.*

[84] Sandeep Tata, Alexandrin Popescul, Marc Najork, Mike Colagrosso, Julian Gibbons, Alan Green, Alexandre Mah, Michael Smith, Divanshu Garg, Cayden Meyer, and Reuben Kan. 2017. Quick access: Building a smart experience for Google Drive. In *Proc. KDD.*

[85] Jaime Teevan, Eytan Adar, Rosie Jones, and Michael A.S. Potts. 2007. Information re-retrieval: Repeat queries in Yahoo's logs. In *Proc. SIGIR.*

[86] Nava Tintarev and Judith Masthoff. 2012. Evaluating the effectiveness of explanations for recommender systems. *UMUAI* 22, 4 (2012).

[87] Nava Tintarev and Judith Masthoff. 2015. Explaining recommendations: Design and evaluation. In *Recommender Systems Handbook.*

[88] Erico Tjoa and Cuntai Guan. 2020. A survey on explainable artificial intelligence (XAI): Toward medical XAI. *IEEE TNNLS* 32, 11 (2020).

[89] Tuan A. Tran, Sven Schwarz, Claudia Niederée, Heiko Maus, and Nattiya Kanhabua. 2016. The forgotten needle in my collections: Task-aware ranking of documents in semantic information space. In *Proc. CHIIR.*

[90] Jesse Vig, Shilad Sen, and John Riedl. 2009. Tagsplanations: Explaining recommendations using tags. In *Proc. IUI.*

[91] Chao Wang and Srinivasan Parthasarathy. 2006. Summarizing itemset patterns using probabilistic models. In *Proc. KDD.*

[92] Fulton Wang and Cynthia Rudin. 2015. Falling rule lists. In *Proc. AISTATS.*

[93] Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. 2017. A Bayesian framework for learning rule sets for interpretable classification. *JMLR* 18, 1 (2017).

[94] Yuhao Wen, Xiaodan Zhu, Sudeepa Roy, and Jun Yang. 2018. Interactive summarization and exploration of top aggregate query answers. In *Proc. VLDB.*

[95] Steve Whittaker. 2011. Personal information management: From information consumption to curation. *ARIST* 45, 1 (2011).

[96] Steve Whittaker, Ofer Bergman, and Paul Clough. 2010. Easy on that trigger dad: A study of long term family photo retrieval. *Personal and Ubiquitous Computing* 14, 1 (2010).

[97] Xuhai Xu, Ahmed Hassan Awadallah, Susan T. Dumais, Farheen Omar, Bogdan Popp, Robert Rounthwaite, and Farnaz Jahanbakhsh. 2020. Understanding user behavior for document recommendation. In *Proc. WWW.*

[98] Xifeng Yan, Hong Cheng, Jiawei Han, and Dong Xin. 2005. Summarizing itemset patterns: A profile-based approach. In *Proc. KDD.*

[99] Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. 2017. Recent advances in document summarization. *KAIS* 53, 2 (2017).

[100] Yongfeng Zhang, Xu Chen, et al. 2020. Explainable recommendation: A survey and new perspectives. *FTIR* 14, 1 (2020).

[101] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proc. SIGIR.*