Nowadays, pervasive computing devices (e.g., laptop computers, mobile phones, IoT devices) largely facilitate our lives. They support billions of users to make decisions by interacting with these devices, e.g., security decisions (i.e., whether to grant data access), shopping decisions, and business decisions. Decision making is a slow judgment process that involves complicated mental models and user efforts in researching, exploration, learning new knowledge, and comparison. Decisions often have to be made out of uncertainty, which means there exists a gap between the user and the knowledge required for making the optimal decision. Two main causes for such gap are: (1) users’ unfamiliarity with the decision domain: decision making often requires the user’s prior knowledge. For instance, in the Android security system, users need to make decisions on whether to grant system permissions (e.g., location, contacts) for each app. If the user is unfamiliar with Android security system, they tend to have difficulty understanding the purpose of permission requests, resulting in the difficulty in making decisions. (2) making decisions in a large database: even if the user is familiar with the decision domain and can easily make decisions over a small number of options, it is still challenging when it comes to making decisions out of a large database, e.g., online shopping. Although systems often support keywords search, users may not know what items are available in the database or their own preferences in advance (i.e., exploratory search). Because of the knowledge gap in (1) and (2), users may find it challenging to make decisions without further assistance from the system.

There exist two major approaches for the system to assist the user’s decision making tasks: first, the system can directly provide automatic decision suggestions so that users can adopt them (e.g., auto completion); second, rather than directly suggesting decisions, the system provides additional information as decision support (e.g., explanations, summarization, and visualization) to bridge the gap. While the first strategy is simpler, it suffers from low explainability and low transparency [1]. More importantly, many times the system is not capable of providing good suggestions, due to the uncertainty in both user intents and the environment. For the example of Android permissions, the integrity of a permission request is context dependent (e.g., when a GPS app requests the location vs. when a camera app requests the location), the system cannot foresee all the contexts and control the access with pre-defined policies (e.g., allowing apps to specify the access with policy language). As a result, the system has to rely on users to make decisions in each case.

Most of my past work focuses on leveraging machine learning/data mining to suggest supporting information to help users with their decision-making tasks, including: (1) suggesting explanations to familiarize users with domain-specific knowledge [2, 3]; (2) summarizing the large database to help users navigate [4, 5, 6]. In addition, I also study how to evaluate the effectiveness of end-user decision making in an exploratory search system [7], and how to improve a recommender system’s performance by leveraging users’ fatigue behaviors [8].

1 Past Work

1.1 Suggesting Explanations for Security Decision Making

The first part of my research focuses on suggesting knowledge to support Android security decision making. Android permission system controls the access to users’ private data (e.g., location, contact list). Every app must request permissions from the user, and the user must allow the permission before an app can get access to the data. However, studies show that Android users often do not understand the purpose of mobile apps’ permission requests, making the decision task difficult [9]. The difficulty comes from the design of Android permission systems. The Android permission system has a hierarchical structure, it consists of 10 permission groups, each group controls one or a few permissions, and each permission controls multiple fine-grained purposes (i.e., the API level). While the app is implemented at the API level, the permission request itself only tells the group level information. As a result, it is unclear whether existing apps provide sufficient support to help users understand the low-level permission purpose and if not, how to improve the decision support.

Examining the Quality of Permission Explanation Sentences. To study the sufficiency of decision support, I examine permission rationales in Android runtime permission systems (Android 6.0). Permission
rationales are texts on mobile interfaces that directly explain the purposes of permission requests. In runtime systems, these rationales are often displayed before or after the permission requests. An example of Android permission rationales is shown in Figure 1. After the runtime model was launched in 2015, more and more apps have included permission rationales, but there remain multiple questions on the sufficiency of decision support: (1) how many apps contain rationales? (2) if an app requests multiple permissions, which permission is more likely explained? (3) are there any incorrect explanations? (4) how specific are the explanations? To answer these questions, I conduct a measurement study [3] on 80K Android apps under the runtime model. I extract the permission rationales from the apk files by leveraging natural language patterns of these rationales. For the above four questions, I find (1) less than 25% apps contain at least one rationale; (2) apps prioritize explaining purposes of frequently requested permissions more than infrequently requested permissions, i.e., apps would more likely explain permissions that are easier to understand; (3) there exist 11.3% incorrect explanations among the apps that request the \texttt{PHONE} permission; and (4) 22%-77% explanations are too general, which basically repeat the permission requests themselves. The results from (1) - (4) imply that Android permission purposes are not sufficiently explained in existing rationales, therefore users need more support for their security decision making.

Assisting Apps to Provide Permission Explanations. Study results in [3] imply the need for better support in users’ security decision making. To assist apps and developers to improve the quality of permission explanations, I study how to recommend candidate explanation sentences for app developers to refer to. The recommender system, CLAP [2], suggests sentences from similar apps’ descriptions using information retrieval techniques. One challenge in this problem is that description sentences from similar apps could not be directly adopted for explaining the current app because they may be irrelevant to the current app’s real purpose. If many of the recommended sentences are irrelevant, app developers would have to spend a long time manually filtering out these sentences. CLAP uses a three-step process to improve the relevance of the recommended sentences: first, CLAP splits sentences into smaller units (so that likely a small unit does not include irrelevant information) and put them in a candidate sentence set to be used later; second, CLAP re-ranks all the candidate sentences by the frequency of the purpose, so the most frequent purposes are ranked to the top; third, CLAP leverages templates to improve the interpretability of recommended sentences. Our experimental results show that more than 80% sentences suggested by our system match the true purpose of the permission requests. In addition, the suggested sentences demonstrate three important characteristics of interpretability: the sentences are diverse, concise (which is useful for displaying on small screens, as mobile users tend to skip long texts), and they define concrete purposes (i.e., in contrast to being too general, a problem we found in existing runtime rationales [3]).

1.2 Interactive Text Data Exploration

The second part of my research studies mining external knowledge for interactive data exploration in a text corpus. When exploring a text corpus, it is helpful to suggest a list of candidate topics to help the user explore the information space and refine her query keywords. For example, the list in Figure 2 suggests topics related to \texttt{deep learning}. The system can further suggest finer-gained topics with a hierarchical structure.

In [5] we study the problem of building a topic hierarchy to support users’ interactive exploration of a text corpus. Existing work focuses on automatically constructing such hierarchies, but one problem with automatic approaches is that the parent-child relations are often erroneous or do not meet the user’s need/context. As a result, we study how to assist users so they can browse the hierarchy while being able to correct the errors at the same time. First, we use [6] to construct the initial hierarchy, then we suggest five editing operations so that the user can use them to correct the errors. One challenge in this problem was how to recompute the tree so that after the user’s editing operations, the unedited part of the tree remain consistent. To solve this challenge, we leverage a robust inference framework named the \texttt{moment-based inference framework}. In [6] we also optimize the time complexity of the state-of-the-art moment-based inference algorithm. Experimental results show that our algorithm reduces the time cost for constructing the hierarchy by orders of magnitude.
1.3 Suggesting Numerical Ranges for e-Commerce Search

The third part of my research studies how to suggest external knowledge to help e-Commerce search. In e-Commerce search, the major challenge for user decision making is that the decision has to be made out of a large catalog, therefore it is unlikely the user could traverse all the potential items. Although many e-Commerce sites support keywords search, it is difficult for the user to input complicated queries with keywords further more, the user often does not know her need in advance, i.e., exploratory search, especially when the decision requires domain knowledge on the item facets, e.g., the electronics department. As a result of users’ difficulty in specifying their need, e-Commerce search engines often suggest a list of item facets to help users to better specify their needs, i.e., faceted search interface. For example, Figure 3 shows the price facet suggestion for TV on Amazon (other facets include brand, screen size, etc.).

Numerical Facets Partition. Despite the popularity of e-Commerce businesses, the faceted search interfaces in many major e-Commerce sites are still suboptimal. One of the most widely unoptimized components is the numerical range facets, e.g., price ranges. The problem with these price ranges is that many e-Commerce engines use the same price ranges for all queries, resulting in heavily unbalanced price distributions: if the search results are expensive (e.g., diamond rings), all results locate in the most expensive range; while if the results are cheap (e.g., greeting cards), all results in the cheapest range. In other words, the partition does not actually help with the exploration, and the user’s efforts on searching remain the same with the partition compared to without. On the other hand, if the ranges are adaptive to different queries, the search results would be more balanced across different ranges, so it requires less efforts from the user to reach the relevant item (the user just have to search within the most relevant range).

By identifying this real problem across top e-Commerce sites, I propose to suggest the numerical ranges that optimize users’ costs (when they using the ranges for browsing). By collaborating with Walmart Labs, I obtain a large-scale real users’ search log in a two-month window. These logs contain users’ query and click records, which define the users’ browsing costs (i.e., the cost equals the rank of the clicked item). With the help of Walmart search log, I design three approaches for optimizing the browsing costs. First, I study to directly optimize the costs, however, we do not know the exact costs before observing the clicks, although we can estimate the costs using probabilistic models, such models are often not accurate enough. As a result, I design a machine learning-based approach by mapping all ranges to a shared parameter space, i.e., the relative proportions of partition points. This step of transformation helps improve the accuracy because the browsing costs are known in the training data. Machine learning proves to improve the results, and non-linear methods (e.g., decision tree) are helpful for further improvements. However, the optimization in the machine learning approach has huge computational cost. To this end, I further optimize the computational cost by optimizing the upper bound instead of the original objective function. Experimental results show that compared with the baseline (i.e., evenly partitioning the ranges), decision tree saves up to 21% of users’ browsing costs.

2 Ongoing Work: Natural Language Programming Interface

One major challenge in user decision making is the lack of support in natural language interfaces/semantic parsing. Semantic parsing allows users to easily talk to intelligent personal assistants, e.g., “find the best rated restaurant for family within driving distance”. In data analytics, semantic parsing also allows a novice data analyst more conveniently query databases without having to master complicated grammars. Such support is important due to the tremendous growth of data industry and the prevalence of data analytics jobs. Semantic parsing is especially useful for voice input, which will outperform traditional typing by 2020.

Unfortunately, existing work in semantic parsing are either non-generalizable or not precise enough. Semantic parsing has been studied for several decades. In the past, the task was studied mainly within one domain. For example, by exhaustively enumerating possible questions for flight tickets search, one can train a semantic parser that performs well on users’ flight-tickets-related questions. However, such results cannot generalize to

---

1 User query log on Walmart.com also contains few complicated queries; most queries contain less than 5 words.
new domains. In the last two years, multiple cross-domain datasets have been created for training semantic parsers from scratch, i.e., with zero labeled data in a new domain. Specifically, state-of-the-art techniques have achieved 86% accuracy on a cross-domain task (WikiSQL dataset). However, the target SQL queries in WikiSQL is criticized as oversimple and does not reflect the performance on realistic scenarios. A newly released dataset named Spider solves this problem by including realistic/complicated target SQL queries. I am working on jointly learning from the two datasets to train a semantic parser which is generalizable enough while also understands complex grammars.

3 Future Work

My research agenda is to computationally improve user experience in users’ decision making tasks with machine learning/data mining/information retrieval techniques. In future, I plan to work on both of the aforementioned approaches for decision assistance, i.e., (1) directly suggestion decisions and (2) suggesting external knowledge for decision support. In particular, I am interested in the following domains: (1) e-Commerce search: how to assist users to quickly find items, i.e., helping them make near-optimal decisions within a short time of exploration? (2) security: how to improve user understanding of security operations? (3) programming: how to reduce programmer’s efforts by (semi-)automatically synthesizing code, documentation, etc.? (4) education/peer review: how to reduce teacher/reviewers' grading/reviewing efforts with (semi-)automatic grading/peer review? Section 3.1-Section 3.4 explain each direction in more details.

3.1 e-Commerce Search

As an e-Commerce business/catalog grows larger, user exploration also becomes more difficult: a majority of Walmart’s catalog have never been purchased or viewed before. When the search engine uses machine learning to train its ranking algorithm, the ranking becomes even more biased towards the most popular items and user exploration becomes even harder, which leads to suboptimal decisions. How to assist users to more thoroughly explore the information space while reducing their efforts in this process? For this problem, I plan to study an intelligent-agent approach. The agent explicitly queries the user’s decision rules: (1) the agent pro-actively asks questions on the facets to learn the user’s preference; (2) the agent crawls text data from multiple resources (e.g., user reviews) to support the user’s decision choices; (3) the agent explains decision rules using an economic model, e.g., “would you pay 50$ more to get a cashmere sweater?”. By explicitly modeling the user’s decision rules, the agent can better understand the user’s need and help the user explore less popular items which are potentially the optimal options. I plan to continue working with Walmart Labs to explore these topics.

3.2 Security Interaction

My existing work on mobile security interaction studies an important problem in data transparency. Policy makers and media have emphasized the importance of users’ rights to have full control over their personal data. In 2018, the Facebook Cambridge Analytica scandal came out and the European Union’s data protection regulation (GDPR) came into effects. GDPR requires companies to make sure data is processed in a lawful manner, and users must understand how their data is processed. While my existing work helps users to understand the purpose of data usage, one limitation is that it is not secure enough. It cannot detect the case where an app claims a wrong purpose. If the developer directly adopts a wrong purpose, it causes security vulnerability in the app. One direction to strengthen the security is by detecting the real purpose of an app with program analysis. A recent work by security researchers on malware detection with convolutional neural network on control flow graph shows promise of such direction. Besides extending my existing work on mobile security, I also plan to explore other domains including IoT security.

3.3 Natural Language Programming Interface

Existing work on semantic parsing/program synthesis focuses on learning from benchmark datasets. However, because most of such benchmarks are collected from crowdsourcing on Amazon Mechanical Turk, they tend to suffer from a common problem of being too “toy” (i.e., the goldstandard does not solve real users’ needs even if the synthesis is correct, e.g., question = “print the value of a”, programming language = “print a;”). In contrast, questions/answers from Stack Overflow more realistically resemble users’ actual needs, but existing synthesis techniques are not robust enough for learning useful information from such noisy data sources. Below are two questions I plan to explore: (1) can massive but noisy Stack Overflow data help with the learning? For example, by combining the learning with program analysis? (2) can synthesis help reduce user efforts in
searching for the correct answer? For example, by building an intelligent agent which replaces user efforts in scanning and removing obviously wrong answers.

### 3.4 Other Decision Support

In future, I also plan to explore decision support on other tasks. For example, OpenReview.net consists of peer reviewer decisions for all submissions to ICLR conference. How can we leverage the peer review data to support authors/reviewers’ decision making tasks? We plan to study citation recommendations to help authors identify controversial citations and improve the quality of their manuscripts. In the long term, I am also interested in exploring more topics in security, software engineering, healthcare and education.

### References


