Nowadays, pervasive computing devices (e.g., laptop computers, mobile phones, IoT devices) largely facilitate our lives. They support billions of end users to make decisions by interacting with these devices, e.g., installing software, shopping, and booking flights. Here the definition of a decision is one where an end user has to choose between multiple options, and different choices can result in a difference in the user's personal benefit, e.g., money, time, or security. Decisions often have to be made out of uncertainty, because there exists a knowledge gap between an end user and the decision-making environment. Such gap can come from the following two causes. (1) **End user's unfamiliarity with the decision domain**. One reason of the unfamiliarity is that the decision domain is too technical for the end user to understand. For instance, in the Android security system, the user needs to make decisions on whether to grant system permissions (e.g., location, contacts) for each app. However, the user is unfamiliar with the system operations (e.g., permissions, API calls, third-party libraries). As a result, the user often has difficulty understanding the purpose of permission requests, resulting in the difficulty to make permission decisions. (2) **Information overloading in the system**. Even if the end user is familiar with the decision domain and can make the perfect decision over a small number of options, the decision making can still be challenging when the decision has to be made out of hundreds or thousands of options. In the latter case, the challenge lies in how to find the *optimal* option without traversing every option. In other words, there exists a knowledge gap between the user and the availability of better options. Because of the knowledge gap in (1) and (2), the user may find it challenging to make decisions without any assistance from the system.

There are two main strategies for the system to assist the user's decision-making tasks: first, the system can directly provide automatic decision suggestions so that the user can adopt (e.g., auto completion); second, the system does not directly suggest decisions, but provides *decision-supportive tools* (e.g., explanation, summarization, and visualization) to help the user better understand and manage the decision-making tasks. While the first strategy is simpler, it suffers from low explainability and low transparency [1]. More importantly, often the time 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 or control the access with pre-defined policies (e.g., allowing apps to specify the access with a policy language). For this reason, the system has to rely on the user to make decisions in each case.

My research follows the second strategy to **develop decision-supportive tools that help the user with the decision-making tasks**. Most of my existing work falls into the following two categories: (1) *providing explanations to familiarize the user with domain-specific knowledge* [2, 3]; (2) *summarizing a large database to help the user navigate* [4, 5, 6]. In addition, I also studied how to evaluate the effectiveness of end-user decision making in an exploratory search system [7], and how to improve a ranking system's performance by leveraging the end user's exploratory behaviors in decision makings [8].

Providing Explanations to Familiarize Users with Domain-Specific Knowledge

The first part of my research focuses on explaining domain-specific knowledge to end users to bridge the gap in their decision making. More specifically, I focus on the case of decision making for Android security. Android users often do not understand the purpose of mobile apps' permission requests, making it difficult for the users to choose between allowing or denying the permission [9]. Android permissions control the access to users' sensitive information. While the apps request access from the API level, a user can only see the request from the permission level. As a result, the same permission request may come from many different contexts of API usages, but from the user's perspective, the request looks the same. Previous studies show that it often confuses

the user whether a request is benign or *over-privileged*. To justify their permission usage and bring transparency to the permission request, many apps provide a natural-language sentence explaining the purpose of the request, e.g., *To attach media, we need access to your gallery.*

Studying the Quality of Permission-Explanation Sentences. Because permissions are requested by app developers in the first place, it is not a major challenge for the developers to provide a sentence stating the purpose for the request. The real challenge lies in how to *clearly* explain such permissions to an end user, who is less familiar with Android permissions and APIs. For instance, if the explanation is too abstract, the user may not learn anything new, but if the explanation is too specific, e.g., with many technical details, it can also be difficult for the user to process the technical information. To study the quality of permission-explanation sentences and whether they clearly explain the permission purpose, I conducted a large-scale measurement study on the existing runtime permission rationales in Android apps [3] (an example of such sentences is in Figure 1, explaining why the app requested the permission for reading the user's external storage). I designed a comprehensive methodology for measuring multiple aspects of an explanation's quality: overall frequency, categorical distributions, correctness, and specificity. For example, Android developers' guidelines suggest apps to explain permissions that do not look straightforward to users, e.g., a camera app uses users' location permission (vs. when it uses the camera permission). To evaluate whether apps are following such guidelines, I examined the statistical correlation between an app's permission-explanation behavior and the straightforwardness of such permission requests. The results showed that a majority of the studied apps were not following such guideline; in addition, many explanations were mistaken, and a large proportion did not explain any substantial purposes. In summary, the quality of explanations needs to be much improved.

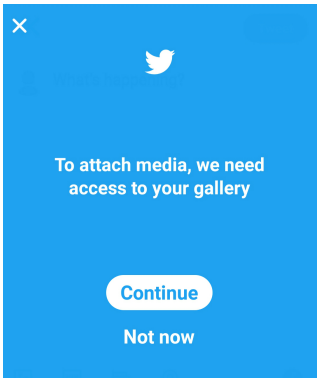

Figure 1: An app-provided explanation for the purpose of permission request

Assisting Apps to Provide Permission Explanations. The low quality of permission explanations in existing Android runtime rationales indicates the challenge for app developers to clearly explain permission purposes to users. To assist developers to improve the quality of their explanations, I built a recommender system named CLAP [2] that recommends candidate sentences for the app developers to refer to. CLAP selects the recommended sentences by conducting information retrieval techniques on the descriptions from the most similar apps. One challenge in this approach was that description sentences from similar apps could not be directly adopted for explaining the current app because they may (partially) describe a different purpose than the current app's purpose. If many of the recommended sentences were irrelevant, app developers would have to spend a long time manually filtering out these sentences. CLAP used a three-step process to improve the relevance of the recommended sentences: first, CLAP split sentences into smaller units and put all the resulting sentences into a candidate set, so that some sentences in the candidate set may contain less irrelevant information than the original sentence; second, among all the sentences, CLAP ranked candidate sentences by their semantic popularity, i.e., how frequently the sentence's purpose is mentioned among similar apps; third, CLAP leveraged a few templates to improve the interpretability of recommended sentences. Our experimental results showed that at least 80% of the top-5 recommended sentences matched the ground-truth purpose of the permission request. In addition, our case-study results showed that the top-ranked sentences provided a diverse choices of wording, and these sentences were concise but they explained concrete purposes.

Providing Summarization to Assist Decision Making on Large Databases

Even if a user is familiar with the decision domain and has little difficulty making binary decisions, the decision-making task is still challenging when the decision has to be made from hundreds or thousands of options, because the user does not know whether there exists a better option among the options that the user has not visited. Such case is called *exploratory search*, where the user is looking for the best option (exploitation) and learning new information about the available options (exploration) at the same time. In those cases, we can provide a summarization of items in the database (e.g., Figures 2 and 3) to help the user better understand what is available and help the user more efficiently explore the database.

Numerical Facets Partition. Numerical facets are ubiquitous in end-user application domains, such as price (shopping and travel), distance and rating (local), ages (dating), calories (fitness), and income (jobs). They also widely exist in the domains of software engineering and security, e.g., lines of code (LOC), votes for forum questions, the severity grade of a security vulnerability, and the grade of a student. Compared with other facets (i.e., categorical facets), numerical facets provide a succinct and interpretable summarization for the data distribution. For example, the price ranges in Figure 2 tell users that all TVs can be briefly categorized into the five grades, and a large proportion of TVs are under 500$. In [4] I studied the problem of recommending numerical-facet partition to help users with exploratory e-Commerce search. The main challenges in this problem were of two folds: first, how to quantitatively evaluate the performance of the recommended numerical facets; second, how to search for the numerical facets that optimize such evaluation metric. For the first challenge, I proposed to use users' browsing cost (while using the facets as expansion) as the evaluation metric. With such defined metric, a follow-up challenge was on how to optimize the browsing cost and yet the cost was unknown during testing. Existing work could be used to estimate the cost, but the accuracy was low. Instead of adopting existing work, I identified that one key step could help solve the problem: a transformation on the math representation of the parameter space. Moreover, some math proofs showed that after this step of transformation, the objective function had an (approximated) upper bound, and the computational time complexity for optimizing the upper bound was largely reduced when compared with that for optimizing the original objective function [4]. Furthermore, I developed an advanced model that was more adaptive to different contexts (e.g., query text) based on the regression tree algorithm. Experimental results showed that users can save 25% of efforts when using my system to navigate compared with when

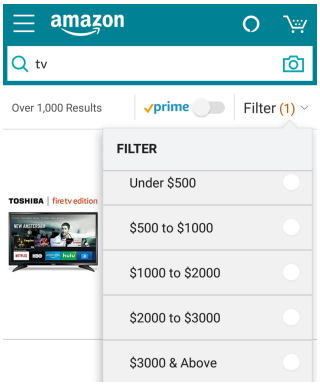

Figure 2: A numerical facet navigational system for e-Commerce search

Interactive Construction of Hierarchical Topics. In literature search, when a user does not know what topics are available in the database, the user may want to explore the topics using a topic classification system. For example, Figure 3 shows a flat topic classification system, where the listed topics can help the user refine her/his query *deep learning*. On the other hand, a hierarchical classification system can further suggest the finer topics within a topic. In [10] we studied the problem of building a hierarchical topic classification system out of a text corpus by *interacting* with an end user. Although existing work studied how to *automatically* construct such hierarchies, the parent-child relations in automatically constructed hierarchies may often be error-prone. As a result, we

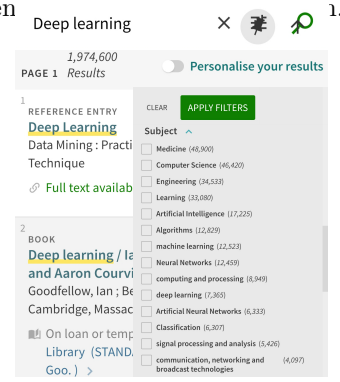

Figure 3: Topic Navigation

studied how to assist the end user so she/he can browse the hierarchy while also being able to correct the errors at the same time. First, we used [6] to construct the initial hierarchy, and then we provided 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 remains consistent. To solve this challenge, we leveraged a robust inference framework named the *moment-based inference framework*. In [6] we also optimized the time complexity of the state-of-the-art moment-based inference algorithm. Experimental results showed that our algorithm outperformed the existing algorithm, being orders of magnitude faster in the construction speed.

Other Work on Assisting End-User Decision Making

Evaluating a System's Effectiveness for Exploratory Search. Most of existing work on information retrieval leverages the Cranfield testing methodology, which evaluates a system's ranking performance on a fixed set of queries and document lists. However, this methodology cannot be used to evaluate today's search engines, because today users more frequently perform exploratory search, e.g., Google image search now supports exploratory search with tags. To bridge this gap, we proposed a general framework [7] for evaluating the effectiveness of an exploratory search system. To solve the challenges in evaluating a mixture of user operations (e.g., click on document, click on tags, query reformulation) in exploratory search, we jointly modeled these operations using a cost-benefit framework. We tested the integrity of our evaluation framework by comparing its results with the evaluation results from a real user study. Such experiments showed that our framework preserved the consistency of the ordering results between information retrieval systems compared to that from the real user experiment.

End User's Exploratory Behavior on Bing News Recommender System. During my internship at Microsoft Research, I worked on Bing news recommendation [8]. In this internship, I got the chance of analyzing the user behaviors on an industry-scale user-click log. I found that users were much more prone to clicking on a news article the first time they saw it; such observation can be explained by users' preference of exploring new information. The click-through rate of the same news article significantly decreased over time, regardless of whether it had been clicked before. In addition, for users who frequently interact with the system, the click-through rates decreased even faster. Based on such observations, we discussed the user-fatigue phenomenon in a news recommender system, designed and tested multiple fatigue features under Bing's infrastructure, and experimental results showed that these features could improve the system performance by 15% overall and 34% for frequent users.

Future Work

My existing work had a theme on how to explain information (data distribution and domain-specific knowledge) to support end users' decision-making tasks. With the new data transparency regulation (GDPR), the explainability of a system has become one of the most critical problems today. In future work, I plan to study how to further improve the explainability of systems to support end-user decision making. My future work includes the following three short-term directions, and two long-term directions.

Improving Permission Explanations in Mobile-Security Systems. I plan to further improve my existing work on explaining mobile-permission purposes in the following directions. First, *explaining permissions by leveraging program analysis*. Natural-language explanations help end users understand the functionality behind security risks. However, if the explanation is inconsistent with the apps' true functionality, attackers

can exploit such inconsistency to fool users into accepting the permission. I plan to improve the security of CLAP by examining the consistency between natural-language statements and the code using static analysis. Second, *recommending the right time of permission explanations.* My previous work mostly focused on what to explain [3] and how to explain [2] for app permissions. On the other hand, timing (when to explain) is another question that app developers need to decide under the runtime permission system. Explaining at the right time means that the explanation is consistent with the context, while also not being redundant or verbose. I plan to study assisting developers to decide the right timing of permission explanations, e.g., by learning from when similar apps explain their permissions. Third, *studying the causal relations in permission explanations.* Even if an explanation sentence explains the purpose, the relation between the cause (achieving the functionality) and the effect (using the permission) may not look straightforward to users, e.g., an alarm app justifies its location permission using "*we need to use your location to send you weather report*", but from the user's perspective, it may not be clear on why an alarm app needs the functionality of weather report. As a result, an app can provide further assistance to strengthen the causal reasoning. For example, such evidence can be statistics on how many similar apps use the same functionality.

Providing Explanations in Other Domains (e.g., Education and IoT). My existing work mostly focuses on mobile security and web search. In future work, I plan to study how explanations can help with users' decision making in other domains. First, *explaining students errors in education.* Because students are beginners in their technical domain, it is often expected that they have insufficient expertise in the tasks that they work on. For instance, in an introductory programming course, students may experience compilation errors, and beginner programmers may not be able to understand such errors very well. As a result, it is helpful to provide an explanation sentence to help the students better understand the compilation errors. For example, the explanation may state whether an index out of range error is due to the upper bound or the lower bound. Second, *explaining IoT permission purposes.* Security vulnerabilities on IoT devices can lead to physical attack, and more sensitive private information are used (e.g., when the user is eating breakfast). As a result, IoT users often cautiously examine the permission purpose. While IoT permissions are mostly similar to mobile permissions, there exist major new challenges for explaining IoT permissions. For example, IFTTT applets' descriptions are much shorter than mobile apps' descriptions, therefore calling for the new explanation needs to bridge the semantic gap between the applets' descriptions and permissions.

Supporting End-User Decision Making with Conversational Question-Answering Systems. A limitation of my existing work is that a user can only *passively* receive the explanation, but cannot *actively* ask questions or express her/his confusion on the decision-making task. In future work, I plan to study bridging this gap by supporting a conversational question-answering system between the users and the system. For instance, for mobile security, the users may ask the following questions "*is this app uploading my contact lists to a server outside of US?*", "*can you recommend an alternative app with the same functionality but not requesting my location?*" The system then answers the user's questions by leveraging a combination of techniques on program analysis, data mining, information retrieval, and natural-language generation. Furthermore, because the users are unfamiliar with the system, they may not even know what/how to ask in their questions, and the system can assist them to ask questions by recommending what is the most likely uncertain question from the users' perspective.

***Long-term*: Studying Users' Cognitive Bias in Decision Making**. End users' security-decision making may be subject to cognitive bias. In security-decision making, the users may trust apps from big companies more than those from small companies, because the users see those apps more often. In e-Commerce exploratory search, a user may develop bias on what facets are related to "*better quality*", e.g., if the user sees many Samsung

TVs in the search results, the user may quickly jump into the conclusion that the user needs a Samsung TV too, although better options are available. In the long term, I am interested in studying whether it is possible to provide decision-supportive information to mitigate such bias. Can we mathematically model such biases? Can we systematically help with users' decision making by showing surprising results that contradicts with their biases?

***Long-term*: Improving Transparency in Security-Decision Making**. Since the beginning of the Android permission system, the users have always been the ones to make the decisions on the apps' access control. With fine-grained decisions in Android 6.0, Android has provided developers with a tool named Android vitals, which can help measure the users' denial rates. However, it is unclear whether such denial rates can indicate the security of the app. If they can, then can we predict the denial rates to help developers with testing? If they cannot, then the current approach of asking users to make security decisions may not be secure enough.

In summary, my research has been on developing decision-supportive tools that help users with their decision-making tasks. In future work, I am excited to continue working on the preceding future directions, and with my interdisciplinary background, I am interested in collaborating with researchers from software engineering/programming language, security, data mining, information retrieval, HCI, machine learning/AI, social science, and other areas.

References

[1] B. Goodman and S. Flaxman, "European Union regulations on algorithmic decision-making and a "right to explanation"," *ICML Workshop on Human Interpretability in Machine Learning (WHI 2016)*, 2016.

[2] X. Liu, Y. Leng, W. Yang, C. Zhai, and T. Xie, "Mining Android app descriptions for permission requirements recommendation," in *Proceedings of the International Requirements Engineering Conference (RE 2018)*, 2018, pp. 147–158.

[3] X. Liu, Y. Leng, W. Yang, W. Wang, C. Zhai, and T. Xie, "A large-scale empirical study on Android Runtime-Permission Rationale Messages," in *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2018)*, 2018, pp. 137–146.

[4] X. Liu, C. Zhai, W. Han, and O. Gungor, "Numerical facet range partition: Evaluation metric and methods," in *Proceedings of the International Conference on World Wide Web Companion (WWW 2017)*, 2017, pp. 662–671.

[5] C. Wang, X. Liu, Y. Song, and J. Han, "Towards interactive construction of topical hierarchy: A recursive tensor decomposition approach," in *Proceedings of the SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2015)*, 2015, pp. 1225–1234.

[6] ——, "Scalable moment-based inference for latent Dirichlet allocation," in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD 2014)*, 2014, pp. 290–305.

[7] Y. Zhang, X. Liu, and C. Zhai, "Information retrieval evaluation as search simulation: A general formal framework for IR evaluation," in *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR 2017)*, 2017, pp. 193–200.

[8] H. Ma, X. Liu, and Z. Shen, "User fatigue in online news recommendation," in *Proceedings of the International Conference on World Wide Web (WWW 2016)*, 2016, pp. 1363–1372.

[9] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android permissions: User attention, comprehension, and behavior," in *Proceedings of the Symposium on Usable Privacy and Security (SOUPS 2012)*, 2012, pp. 3–14.

[10] D. Li, W. Lam, W. Yang, Z. Wu, X. Xiao, and T. Xie, "Towards privacy-preserving mobile apps: A balancing act," in *Proceedings of Hot Topics in Science of Security Symposium and Bootcamp (HotSoS 2018)*, 2018.