My research interests have been on solving users’ “real pains” in their interactions with computers with data-driven/machine learning approaches. Despite the development of computer techniques, users today still face many difficulties when interacting with computers. Take online shopping for example. Statistics show that the return rate of online shopping (30%) is 3.4 times higher than in-store shopping. Such difference indicates the suboptimal experience in online shopping in general, and the difficulty in searching for items online. Generally speaking, three factors can cause difficulties in users’ interactive tasks. First, the difficulty for systems to understand natural language. It is difficult for users to "talk" to the computer the same way as she talks to a human due to the difficulty of natural language understanding by computer systems. Second, the limitations in user’s manual efforts. Since users are human, there exist constraints in the time and efforts users can spend on optimizing the results, and users often quickly satisfy to acceptable (yet suboptimal) results to avoid the efforts for optimization. Third, users’ knowledge gap in new domains. When being unfamiliar with a new domain, users do not know how to make decisions without knowing the background knowledge first. In online shopping, for example, between a TV sales person buying TV and a novice user buying TV, the latter needs more time learning background knowledge (e.g., what does "Plasma" mean and why she prefers it, what are the new features in 2019, etc.).

Machine learning has achieved great success in assisting human-computer interactions in the web search and recommender system domain. For example, by leveraging users’ contextual information (e.g., location) as an input feature for machine learning, the system can learn how to precisely place targeted ads that meet users’ need. The recommendation algorithms are trained on users click logs, and users will likely make similar clicks under the similar situations, e.g., users who are at birth centers are likely new parents, so there is a high chance they buy baby products.

Given the success in search and recommender system, can machine learning also help with other interactive tasks where users have “real pains”? Besides online shopping, users also frequently perform many other difficult tasks in life or in work. Many of such tasks require users’ domain expertise thus take long time for novice users to investigate. For example, software development, medical analysis, and grading. To reduce user efforts in these tasks, one idea is to automate user efforts with an intelligent agent. Such an agent needs to work with human collaboratively instead of completely replacing human’s role, because on the one hand, state-of-the-art AI techniques cannot yet replace human in many tasks, on the other hand, replacing human would cause security (e.g., in software development) and explainability problems. For example, we are still at an early stage for intelligent agents to automate software development (e.g., the idea of software 2.0).

Figure 1 shows a general framework I envision for intelligent agents to work collaboratively with human and target the three challenges mentioned above. First, the framework provides a natural language interface for users to more conveniently specify their intents. In software development, for example, users can specify the programming intent in natural language, e.g., “sort the dictionary by keys”, the interface can talk back to actively question users’ unspecified intent, e.g., “which dictionary do you want to sort?” . Second, the framework computationally minimizes user efforts before reaching the optimal result. It should also support automation or partial automation to reduce user efforts. Third, the framework bridges users’ knowledge gap by providing background knowledge, e.g., natural language explanations or structured knowledge.

1 Existing Work

My existing work has looked into two interactive tasks (for users): e-Commerce search (Section 1.1) and mobile permission requests (Section 1.2). For e-Commerce search, I study how to learn from users’ search log to optimize users’ browsing cost in the search. For mobile permission requests, I study how to bridge users’ knowledge gap by mining natural language explanations from Google Playstore data.

1.1 e-Commerce search: Learning from user search log to optimize user efforts in faceted search
Faceted search interfaces are filters in e-Commerce websites, e.g., departments, brands, price ranges (an example of faceted search interface is in Figure 2). By selecting from the filters, users can define queries with exact constraints (e.g., `SELECT * FROM tv WHERE price < $500`). Moreover, faceted interfaces bridge users’ knowledge gap, e.g., the user gets to know the brands knowledge by viewing the brands facet (which helps with the decision making). Further more, faceted search interfaces are critical for assisting users’ browsing when the keywords search fails. There exist many complicated intents where it is difficult for users to find the keywords equivalents, such as price-related intents (e.g., “Help me find TVs that are less expensive than 400$ and more expensive than 300$.”). On the other hand, natural language queries like this intent cannot yet be understood by state-of-the-art engines in natural language understanding (e.g., Google shopping).

In our WWW 2017 paper, I study how to use machine learning to automatically suggest numerical facet ranges (e.g., price ranges in Figure 2) while optimizing users’ browsing cost with such ranges (i.e., minimizing the time cost for finding the relevant item). By leveraging users’ search log in Walmart e-Commerce’s search engine, I design three algorithms for the optimization, including one that directly optimize the objective cost and two that indirectly optimize it using machine learning. The result shows that machine learning significantly reduces users’ browsing cost (5 items or 25% less items to browse), and the indirect machine learning approach is more effective than the direct optimization approach. The optimization framework is general and it can be extended to tasks other than optimizing the numerical facets. Multiple directions of future work are discussed in Section 2.

1.2 Mobile permission requests: Mining Google Playstore data to explain security knowledge

Mobile penetration rates have kept growing in the recent years. Mobile devices are so prevalent (over 70% world population own a phone, over 67% Americans own a smart phone) that they start to dominate human-computer interaction, with users spending more time on phones than on computers. On mobile platforms, one of the most common interactive activities is security permission requests. All the apps have to request permissions to access users’ private data (e.g., user location, gallery), and users must allow their requests before the apps can proceed and access users’ data. One example of mobile permission request interactions is shown in Figure 3, where the app Twitter requests access to users’ storage data. In Android security system, each permission controls various different purposes, e.g., an app may use the location for GPS tracking or for advertisement. Therefore, users cannot exactly infer the fine-grained permission purpose from just the permission request (like in Figure 3). Moreover, the integrity of mobile privacy depends on the context, e.g., if two apps both track users’ GPS location, the first is a GPS app while the second is a psychological test app, the user tends to find the second app more invasive than the first one.

As a result, it is important for legitimate apps to explain the fine-grained purpose to reduce users’ security concerns. In fact, an existing user study shows that users do not understand the purpose of permission requests 1/3 of the time. Users also frequently complain about mobile permission requests being invasive and confusing on social media, mobile forums and other platforms.

To assist such difficulty in mobile users’ interaction, I look into two research questions. First, are mobile permission purposes already sufficiently explained in existing apps? Second, if they are not sufficiently explained, how to assist the explanations? Both problems can be addressed with data-driven approaches. For the first question, I extract explanation sentences from the apps’ apk files using NLP techniques. I measure four aspects of “sufficiency” of explanations (VL/HCC 2018), and the study results imply that existing apps’ explanations were not sufficient (by the time of the study). As a result, I propose to help existing apps explain their permission purposes with a recommender system approach. The recommender system suggests candidate explanation sentences for app developers to refer to. The candidate sentences are mined from the massive app description data from Google Playstore. The major technical challenge here is how to find an explanation that matches the current app’s purpose without knowing it. To solve such challenge, the recommender system leverages collaborative filtering (i.e., learning from explanations by the most similar apps) and truth finding (i.e., ranking
sentences by the popularity of the purpose). Our manual judgmental results show that the system can recommend relevant explanations in 80% cases. Our further investigation also shows good interpretability of the recommended sentences, which is important in user interaction.

2 Ongoing and Future Work

This section briefly introduces my ongoing work in NL programming interface (Section 2.1) and future work (Section 2.2-2.4).

2.1 Programming: Learning natural language to programming language synthesis

Machine learning techniques for natural language programming interface (semantic parsing) has received much attention in recent years. Semantic parsing is the task of translating a natural language sentence into a formal logical form which can be executed by a computer system (e.g., SQL query). Such task can help with users’ programming activities, because a novice user who does not understand the target grammar tends to find it easier to specify the need using natural language. For example, the fact that users ask “how to” questions on Stack Overflow (e.g., “How to join two tables?”) shows the large gap in user’s domain knowledge, i.e., they can neither write the query from scratch nor find an answer from Google.

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 understands the user’s flight-tickets-related questions well. 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. I am working on synthesizing SQL queries using a newly released dataset named Spider. The spider dataset is also cross-domain, but it also consists of realistic/complicated target SQL queries. I am working on jointly learning from the two datasets to combine the domain breadth of the first task and the complexity of target language in the second task.

2.2 Future work in e-Commerce search

Despite the development of e-Commerce websites, users still find it difficult to search for the optimal shopping decision, especially when the domain is new or when the catalog is huge such as www.taobao.com. A general problem which subsumes my WWW 2017 work is, how can an e-Commerce engine and a user collaboratively find an item that is optimal for the user while reducing users’ searching/browsing efforts in this process? In other words, how can the search engine know almost surely that there does not exist a better option among items that are not visited? The difficulty in this problem lies in that the search engine does not know users’ exact preference, and the user does not know what else is available (plus the aforementioned three difficulties). Other factors that complicate this problem includes users’ learning effects and search engine failure. One potential direction is to formally model this problem as a two player game and use it to compare and optimize different search strategies (e.g., how to reach the “almost surely visited” state with the fastest speed?)

2.3 Future work in mobile security interaction

My existing work on mobile security interaction addresses the important problem of 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. One limitation in my existing work is that it is not secure enough. It cannot detect the case where an app claims a wrong purpose. 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.
2.4 Future work in 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 represent users’ real pains, an extreme example is natural language = “print the value of a”, target = “\texttt{print a;}”). In contrast, questions/answers from Stack Overflow more realistically resemble users’ actual needs, but existing synthesis techniques cannot yet learn useful information from such noisy data sources. One question is, can massive but noisy Stack Overflow data help with the learning? Another question is, 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.