Personalizing users' experience and the ability to perform complex tasks on smart devices and environments such as smart speakers and smart homes are changing the way people are doing their daily tasks. Checking the weather and planning to visit a museum is as simple as asking your smart speaker at home to read out loud the weather condition and commanding the Intelligent Assistant integrated with the smart speaker to book a ticket to visit the museum. To improve user experience in physical spaces such as smart homes, museums, and cities while performing their daily tasks, effective modeling of users interacting with smart devices is required. The overall goal of this thesis is to improve users' experience in physical spaces such as smart cities and environments by modeling user interactions with smart devices. Smart devices hold the promise to bring the powerful tools of the online world into the physical world, and our results highlight similarities and differences with user interactions in traditional search and recommendation settings, and help promote the user experience while interacting with smart devices.

ISBN 978-90-821695-2-2

Modeling Users Interacting with Smart Devices

Seyyed Hadi Hashemi

Modeling Users Interacting with Smart Devices

Seyyed Hadi Hashemi

Modeling Users Interacting with Smart Devices

ILLC Dissertation Series DS-2021-03

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Science Park 107
1098 XG Amsterdam
phone: +31-20-525 6051
e-mail: illc@uva.nl
homepage: http://www.illc.uva.nl/

SIKS Dissertation Series 2021-03.
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

The research for publication of this doctoral thesis received financial assistance in part from the European Community's FP7 (project meSch, grant # 600851), see http://www.mesch-project.eu/. Parts of the research for publication of this doctoral thesis received financial assistance from internships supported by Microsoft Inc. Views expressed in this thesis are not necessarily shared or endorsed by those funding the research.

Cover by Mostafa Dehghani
Published by IR Publications-Amsterdam
ISBN: 978-90-821695-2-2

Modeling Users Interacting with Smart Devices

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. K.I.J. Maex
ten overstaan van een door het College voor Promoties ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op vrijdag 26 februari 2021, te 16.00 uur

door Seyyed Hadi Hashemi
geboren te Sary

Promotiecommissie

Promotores:	dr. ir. J. Kamps	Universiteit van Amsterdam
	prof. dr. W.M.H. Hupperetz	Vrije Universiteit Amsterdam
Overige leden:	dr. I. Zitouni	Google
	prof. dr. C.L.A. Clarke	University of Waterloo
	prof. dr. E. Kanoulas	Universiteit van Amsterdam
	dr. R. Fernandez Rovira	Universiteit van Amsterdam
	prof. dr. L.W.M. Bod	Universiteit van Amsterdam

Faculteit der Geesteswetenschappen

Acknowledgments

Doing a Ph.D. has been the most joyful period of my life, with the opportunity of doing cutting-edge research and contributing to making the world a better place. I will miss the late nights and early mornings hard-work before deadlines, laziness and relaxing after making a deadline, having excitement and stress of opening conference notification emails, all the happiness and sadness after reading the conference notifications, brilliant brainstorming sessions with Jaap, traveling around the world for presenting papers in conferences, learning different cultures, and meeting intellectual people.

When I arrived in the Netherlands for doing a PhD in 2014, I realized how different my years ahead would be. From Jaap's welcome at the airport (which is very rare behavior from university professors in my home country) to funny unstable weather in summer in Amsterdam. Jaap, thanks for giving me the opportunity of doing a PhD under your supervision, helping me to grow, teaching me to manage my time between doing research and product development for the meSch project, being patient, challenging my strange ideas and helping me to make them mature, teaching me how to write high-quality papers, and showing me how to have fun while doing research. You are one of the coolest university professors I have ever seen, and I enjoyed every minute of my PhD (especially, our fun conferences' and EU meSch project trips).

I am very grateful to my promotors, Jaap Kamps and Wim Hupperetz, for accepting me for this PhD position and giving me the freedom to choose interesting problems to work on. Wim, thanks for trusting me and giving me the opportunity to apply my research in the European meSch project and Allard Pierson Museum.

I am also honored to have Imed, Charlie, Evangelos, Raquel, and Rens as my committee members. Thanks for agreeing to be in my PhD committee and generously offering your time.

It was my honor and lucky chance to be invited to co-organize the TREC Contextual Suggestion Track, where I met my great colleagues: Adriel, Julia, and Charlie. Charlie, thanks for all your support and help in my networking and research exposure during conferences.

Many thanks to Hosein and Babak for being my paranymphs and standing by my side in my defense. Hosein, buddy, thanks for all the fun we had during our PhD, all the club-hopping in Seattle, all the adventure we had during conferences and our internship (driving in scary unpaved tunnels and roads in Chile and getting lost there), and all the foosball and billiards we played while running experiments during the internship. Babak, thanks for all the interesting discussions we had about AI and ML Engineering in the last year of my PhD, being a supportive colleague and friend, and a great Iranian cook. You made the most delicious dessert that an Iranian may wish to have at their office.

I want to thank Mostafa for designing the cover of this thesis and Floris for translating the thesis summary to Dutch.

It was a great pleasure and honor for me to work with very talented people at the University of Amsterdam. Beyond work, I enjoyed spending time with you over lunch and coffee breaks, drinks, our social events, playing soccer, and tennis. Thanks a lot: Alex, Avi, Christie, Christophe, Claartje, David, Evangelos, Hugo, Ilya, Inge, Jiyin, Julia, Katya, Liliana, Mahsa, Manos, Merel, Mostafa, Nawal, Nikos, Rosa, Sanna, Tony,

Trond, Xinyi, Yaser, and Zhaochun. I had so many great moments with every single one of you and I wish you all so much success and happiness. I want to especially thank Avi. Avi, I was lucky that you were doing research in Amsterdam when I arrived in the Netherlands. Thanks a lot for all the fun we had in the first 3 months of my PhD. I did not get homesick because of you.

Also, I want to thank my colleagues in the European mesch project. Thank you for the wonderful collaboration in the project: Adriano, Albrecht, Areti, Daniella, Dario, Dick, Elena, Eva, Hub, Ian, Laura, Loraine, Luigina, Mark, Martin, Massimo, Merel, Monika, and Tomas.

During my PhD, I was fortunate to do three internships at Booking.com and Microsoft, which have been resulted in a number of publications. This experience contributed significantly to my academic and career development. Melanie and Pablo were my mentors during my stay at Booking.com. Thank you for giving me freedom in the research and your willingness to help. I truly enjoyed the real-time online experimentation at Booking and evaluating my ideas on many users. Thank you, Melanie, Pablo, Lucas, Ioannis, Yahia, Tolga, and Danil for our fruitful discussions and productive collaborations.

My stay in Microsoft was incredible. I enjoyed every minute of my two internships in Microsoft because of excellent opportunities for research, all the events organized for us, and the beautiful nature in Washington. Imed, thanks for being one of the greatest mentors I have ever had. I learned a lot from you and was impressed by your applied research and leadership skills. Thanks for trusting me and giving me the opportunity of doing two internships in your team. I want to specifically thank Kyle and Ahmed for all their daily supervision during my internships, and all the interesting brainstorming we had. I hope that we will have a chance to work together in the future. I would also like to thank Milad, Ahmed Hassan, and Paul for their support and interesting discussions during the internships. Paul and Milad, it was also quite fun playing football with you guys.

During my internships in Microsoft and presenting my research at conferences, I had the chance of meeting wonderful friends who indirectly helped and motivated me in this journey. Many thanks: Ahmad, Aldo, Aya, Bahareh, Elham, Esfandiar, Faegheh, Hamed, Mehran, Milad, Mohammad, Saghar, and Seamus.

Thanks to all my friends and relatives who made the last few years very enjoyable and indirectly had an influence on my PhD. I want to especially thank Hosein, Keyvan, Kourosh, Mehran, Mojtaba, and Reza. Keyvan, my first friend in Amsterdam, thanks for all the supports and motivating discussions throughout these years. Kourosh, thanks for all the interesting brainstorming for applying my research to the industry, and all the remote support during my PhD. Mehran, thanks for organizing many fun weekends and bringing music to our group. Mojtaba, thanks for being one of my most reliable and caring friends. You have brought lots of fun to our group and made us closer and happier. Reza, thanks for all the great memories from football matches to weekend gatherings. Thanks for organizing sports events and making us healthier in our PhD lives. I have also enjoyed spending time with my other friends in Amsterdam during my PhD. Thank you: Abbas, Ali, Ali Akbar, Akberz, Alireza, Amin, Amir, Amir Hossein, Anna, Arash, Atefeh, Auke, Azarakhsh, Bahram, Behin, Behnam, Behrouz, Bita, David, Dena, Dina, Elahe, Elham, Emad, Erfan, Farhad, Farnoush, Felix, Fereshteh, Golnaz,

Hamed, Hamidreza, Hamraz, Helia, Hendrik, Hiva, Hoda, Irene, Jafar, Jakub, Jon, Kasra, Leila, Mahshid, Mandana, Mansour, Marjan, Martijn, Marzieh, Masoud, Maziar, Mehdi, Mehran, Mehrdad, Mehri, Melissa, Miad, Mina, Minne, Mojdeh, Mozhgan, Nasrin, Navid, Niloofar, Pantea, Pardis, Parisa, Pegah, Pejman, Rokhsareh, Rozita, Saeid, Sajed, Samira, Sepideh, Seringe, Setareh, Shaghayegh, Sima, Simon, Yaser, and Zoheir.

Thanks to my colleagues at ING for interesting discussions about my PhD and their support during my thesis writing period. Thank you: Anand, Floris, Marzieh, Pinar, Shiler, Joost, Chris, Bart, Babak, Ralf, Elvan, Wout, Dimitris, and Mickey.

Special gratitude goes to my family for their unconditional support. Mom, dad, you are always my main driving motivation, and I always admire how you devote every minute of your life to helping and supporting your children. I am sure that I would not be defending my PhD today if I have not had your support and motivation in following science since my childhood. The most difficult part of this PhD was the distance from you and spending less time with you during the past years. Thanks for everything you have done for me. You are always in my heart. I also want to thank my siblings, Maryam, Hesam, and Hosein. Thank you so much for all your motivational speech as an older brother or sister since elementary school. You also contributed to this thesis indirectly. Maryam, without your support, I may have not even got into the AI field. Thank you.

Above all, I would like to thank my wife Hedieh for her love and constant support, for all the late nights and early mornings, for all the sacrifices she made to help me writing the thesis while I was working full-time in the industry, and for keeping me sane over the past few months. But most of all, thank you for being my best friend. I owe you everything.

Seyyed Hadi Hashemi
January, 2021.

Origins and Author Contributions

List of the publications of each chapter and the role of each author is explained as follows:

- **Chapter 2** is based on the following papers:
  - S. H. Hashemi and J. Kamps. Exploiting behavioral user models for point of interest recommendation in smart museums. *New Review of Hypermedia and Multimedia*, 24(3):228–261, 2018 [71]
  - S. H. Hashemi and J. Kamps. Where to go next?: Exploiting behavioral user models in smart environments. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 50–58. ACM, 2017 [68]
  - S. H. Hashemi and J. Kamps. Skip or stay: Users behavior in dealing with onsite information interaction crowd-bias. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, pages 389–392, 2017 [70]
  - S. H. Hashemi, W. Hupperetz, J. Kamps, and M. van der Vaart. Effects of position and time bias on understanding onsite users' behavior. In *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval*, CHIIR '16, pages 277–280. ACM, 2016 [77]

  SHH designed the algorithm, ran the experiments, and did most of the writing; JK contributed to the writing; WH and MV helped in collecting experimental data.

- **Chapter 3** is based on the following paper:
  - S. H. Hashemi, C. L. A. Clarke, J. Kamps, J. Kiseleva, and E. M. Voorhees. Overview of the TREC 2016 contextual suggestion track. In *Proceeding of Text REtrieval Conference (TREC)*, 2016 [76]

  SHH designed the algorithm, ran the experiments, and did most of the writing; JKa helped with the algorithm design and contributed in writing; JKa, CC, JKi, EV contributed in designing the experiments.

- **Chapter 4** is based on the following papers:
  - S. H. Hashemi, C. L. Clarke, A. Dean-Hall, J. Kamps, and J. Kiseleva. An analysis of test collection building in dynamic domains. *Under Submission*, 2020 [82]
  - S. H. Hashemi, C. L. Clarke, A. Dean-Hall, J. Kamps, and J. Kiseleva. An easter egg hunting approach to test collection building in dynamic domains. In *NTCIR-EVIA*, pages 1–8, 2016 [74]
  - S. H. Hashemi, C. L. Clarke, A. Dean-Hall, J. Kamps, and J. Kiseleva. On the reusability of open test collections. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 827–830, 2015 [72]

SHH designed the algorithm, designed and ran the experiments, and did most of the writing; JKa helped with the algorithm design and contributed in writing; AD helped with running the experiments; JKa, CC, AD, JKi, contributed in designing the experiments.

- **Chapter 5** is based on the following paper:

  – S. H. Hashemi, K. Williams, A. El Kholy, I. Zitouni, and P. Crook. Impact of domain and user's learning phase on task and session identification in smart speaker intelligent assistants. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1193–1202, 2018 [80]

  This work was done during an internship at Microsoft in 2017. The task was proposed by IZ; SHH designed the algorithms, ran experiments, and did most of the writing; KW, AK helped with the algorithms design and running experiments. KW, AK, IZ, and PC contributed to the writing.

- **Chapter 6** is based on the following paper:

  – S. H. Hashemi, K. Williams, A. El Kholy, I. Zitouni, and P. Crook. Measuring user satisfaction on smart speaker intelligent assistants using intent sensitive query embeddings. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1183–1192, 2018 [81]

  This work was done during an internship at Microsoft in 2017. The task was proposed by IZ; SHH designed the algorithms, ran experiments, and did most of the writing; KW, AK helped with the algorithms design and running experiments. KW, AK, IZ, and PC contributed to the writing.

Contents

1 Introduction

The last decade witnessed a tremendous interest in creating smart devices and environments helping users finding their information needs in a more personalized and effective way. A smart device is "a context-aware electronic device capable of performing autonomous computing and connecting to other devices wire or wirelessly for data exchange" [146].

One of the main directions towards the creation of smart devices and environments is integrating Intelligent Assistants (IAs) such as Apple Siri, Google Now, Microsoft Cortana and Amazon Alexa in different devices, which has led to the creation of smart devices such as smart speakers [81] or virtually any other appliance including smart microwaves [127]. Another main effort for the creation of smart devices and environments is using the Internet of Things (IoT), which is integrated into physical spaces that have led to the creation of smart environments such as smart museums and smart cities [8].

Each of these smart devices and environments provides device-specific means of user interactions. For example, users interact very differently with smart devices while exploring archaeological objects in a smart museum compared to users interacting with a search engine of the museum to explore archaeological objects. To be more specific, in the smart museum, they might be able to use their RFID tags to unlock the contents of a smart device sharing information about the museum's objects. On the other hand, at the search engine of the museum, they can click on an object to get more information about them.

Since user interaction differs on these different smart devices and a different user interaction leads to a different user behavior, there is a need to study user behavior on these smart devices to provide effective personalized Information Retrieval (IR) systems and improve user experience in physical spaces such as smart museums, cities, and homes. For instance, a user may ask their smart speaker for an advice to visit a museum in Amsterdam, which is the focus of chapters 5 and 6 of this thesis. After confirming the visit to the museum by the user, the user may sit in their smart car, in which the direction to the museum is automatically set in its navigation system. At their arrival, they can visit a smart museum which is using IoT sensor logs and their search engine click-through logs to provide a personalized experience to the user without asking them to explicitly share their preferences. This is the main focus of chapter 2 of this thesis. After the user completed their visit, they may like to have lunch in a restaurant, and use

their mobile phone to receive personalized and contextualized recommendations, which is discussed in chapters 3 and 4 with a focus on creating and maintaining reusable test collections to evaluate contextual suggestion systems.

In this thesis, we study user modeling with an aim of providing a personalized experience for users interacting with smart devices and evaluating users' satisfaction in using smart devices. We first study how to model user behavior to personalize user experience in smart environments. We use behavior modeling to recommend Point Of Interests (POI) in a smart museum. Furthermore, we study POI recommendation in the tourist attraction recommendation domain and a smart city context, in which evaluation of contextual suggestion systems is challenging due to a low degree of reusability of available contextual suggestion test collection. Thus, we organized the Text REtrieval Conference (TREC) contextual suggestion track to create a reusable test collection for the effective evaluation of contextual suggestion systems. As we use dynamic test collections in TREC contextual suggestion track causing test collections' reusability degrade over time, maintaining and improving reusability of test collection is also studied in this thesis.

In addition to model users to provide a personalized experience in smart museums and smart cities, we study user modeling on smart speaker IAs to provide a better user experience in a smart home context. In particular, we study how to identify tasks and sessions on smart speakers with an integrated IA such as Amazon Echo, Google Home, Apple Homepod, and Harmon Kardon Invoke with Microsoft Cortana. We then study how to model users' interaction behavior with smart speakers to predict user satisfaction while fulfilling their information needs in performing a task, which is an implicit signal to improve the effectiveness of IAs.

1.1 Research Outline and Questions

In this thesis, the main aim is to investigate *how to model users interacting with smart devices to improve their experience in the physical space?* To achieve the main aim of this thesis, the thesis addresses improving users experience in physical spaces by (1) modeling users interacting with smart devices in a smart museum to recommend POIs (Chapter 2), (2) creating reusable test collection for offline evaluation of contextual POI recommendation in a smart city context (Chapter 3), (3) maintaining reusability of dynamic test collections for effective evaluation and performance improvement of contextual POI recommendation in a smart city context (Chapter 4), (4) modeling users interacting with smart speaker IAs to identify tasks and sessions from user interaction logs (Chapter 5), and (5) modeling users interacting with smart speaker IAs to predict user satisfaction (Chapter 6). Below, we list the main research question of every chapter.

We first study how to model user behavior based on their interactions with smart devices in a smart museum to provide personalized recommendations of what to see after visiting an initial set of POIs with an aim of improving the user experience at the museum (Chapter 2). The IoT holds the promise to blend real-world and online behavior in principled ways, yet we are only beginning to understand how to effectively exploit insights from the online realm into effective applications in smart environments. We experiment with behavioral user models based on interactions with smart devices

in a museum, and investigate the personalized recommendation of what to see after visiting an initial set of POIs, a key problem in personalizing museum visits or tour guides, and focus on a critical one-shot POI recommendation task—where to go next? We have logged users' onsite physical information interactions during visits in an IoT-augmented museum exhibition at scale. Furthermore, we have collected an even larger set of interaction logs of the search engine of the museum's online collection. In doing this, we answer the following research question:

RQ1 How to model users' information interaction behavior with IoT having an aim of providing a personalized onsite POI recommendation?

To answer this research question, we first study the similarities between users' online digital and onsite physical information interaction behaviors, and build new behavioral user models based on the information interaction behaviors in (1) the physical exhibition space, (2) the online collection, or (3) both. Specifically, we propose a deep neural multi-layer perceptron (MLP) based on explicitly given users' contextual information, and set-based extracted features using users' physical information interaction behaviors and similar users' digital information interaction behaviors.

Next, we study the contextual suggestion task, in which IR systems need to anticipate users' information needs and provide responses relevant to the users' context without the user having to enter an explicit query. To provide a controlled test collection for the IR community for development and evaluation of contextual suggestion systems, we organized the TREC 2016 Contextual Suggestion track that offers a personalized POI recommendation task, in which participants develop systems to give a ranked list of suggestions related to a profile and a context pair available in the tasks' requests provided by the track organizers. Previously, reusability of the contextual suggestion track suffered from using dynamic collections and a shallow pool depth. In this test collection building study for contextual suggestion problem, we answer the following research question:

RQ2 How to create a reusable test collection for the Contextual Suggestion problem?

To answer this research question, the TREC CS web corpus, consisting of a web crawl of the TREC contextual suggestion collection, was made available for the TREC 2016 contextual suggestion track participants. The rich textual descriptions of the web pages make far more information available for each candidate POI in the collection. To create a reusable test collection, a multi-depth pooling approach extending beyond the shallow top 5 pool is used.

As dynamic test collections reusability may degrade over time, we study how we can maintain reusability of dynamic test collections. Search has largely moved to the web and it's many portals and services, yet the dynamic nature of this domain makes it challenging to build reusable test collections. Academic research relies on comparative evaluation using sharable test collection for studying these tasks, and even industrial research having access to online evaluation requires offline evaluation based on editorial judgments for development and analysis. We extensively analyze the test collection building efforts in the TREC 2014 Contextual Suggestion Track, offering a personalized POI recommendation task allowing for either fixed corpus (ClueWeb12) submissions or unrestricted open web submissions. We answer the following research question:

RQ3 Can we build a reusable test collection for a dynamic domain by injecting judged documents into a test collection with sparse judgments?

To answer **RQ3**, we first examine reusability of the original TREC contextual suggestion test collections. We then investigate the expansion of the fixed test collection by inserting open web pages and judgments. Furthermore, we propose a new reusability test for non-pooled runs, called Leave In Uniques (LIU), that is a counterpart of the usual Leave Out Uniques (LOU) for pooled runs.

To improve users experience while interacting with smart speaker IAs in their smart homes, we study how to model users interaction behavior for predicting user satisfaction on smart speaker IAs by identifying IAs tasks and sessions (Chapter 5) and then training a behavioral user model based on user interaction with the smart speaker IAs for user satisfaction prediction (Chapter 6).

We first focus on task and session identification as it is a key element of system evaluation and user behavior modeling in IA systems. However, identifying tasks and sessions for IAs is challenging due to the multi-task nature of IAs and the differences in the ways they are used on different platforms, such as smart-phones, cars, and smart speakers. Furthermore, usage behavior may differ among users depending on their expertise with the system and the tasks they are interested in performing. In this study, we investigate how to identify tasks and sessions in IAs given these differences. In particular, we answer the following research question:

RQ4 What is the impact of the learning curve and task domain on task and session boundaries when interacting with intelligent assistants?

To answer this research question, we analyze data based on the interaction logs of two IAs integrated with smart-speakers. We fit Gaussian Mixture Models to estimate task and session boundaries and show how a model with 3 components models user interactivity time better than a model with 2 components. We then show how session boundaries differ for users depending on whether they are in a learning-phase or not. Finally, we study how user inter-activity times differ depending on the domain of the task that the user is trying to perform.

In the last chapter, by having tasks and sessions identified from users' raw interaction logs with smart speakers, we proceed to a user satisfaction prediction study for users performing a task on smart speakers. IAs are increasingly being used on smart speaker devices, such as Amazon Echo, Google Home, Apple Homepod, and Harmon Kardon Invoke with Cortana. Typically, user satisfaction measurement relies on user interaction signals, such as clicks and scroll movements, to determine if a user was satisfied. However, these signals do not exist for smart speakers, which creates a challenge for user satisfaction evaluation on these devices. We answer the following research question:

RQ5 How to evaluate user satisfaction in Intelligent Assistants based on user queries?

To answer **RQ5**, we propose a new signal, user intent, as a means to measure user satisfaction. We propose to use this signal to model user satisfaction in two ways: 1) by developing intent sensitive word embeddings and then using sequences of these

intent sensitive query representations to measure user satisfaction; 2) by representing a user's interactions with a smart speaker as a sequence of user intents and thus using this sequence to identify user satisfaction.

1.2 Main Contributions

In this section, we list theoretical, algorithmic and empirical contribution of the thesis. For each contribution, we list the chapter from which it originates.

1.2.1 Theoretical contributions

1. Introducing position rank bias, temporal bias and crowd bias in users onsite interaction with smart environments. (Chapter 2)

2. Introducing multi-depth pooling approach extending beyond the shallow top N pool. The multi-depth pooling approach leads to creation of a test collection that provides a more reliable evaluation results in ranks deeper than the traditional pool cut-off. (Chapter 3)

3. Introducing a new reusability test for non-pooled runs, called Leave In Uniques (LIU), that is a counterpart of the usual Leave Out Uniques (LOU) test for pooled runs. (Chapter 4)

4. Introducing sequence of query intent as an implicit signal of user satisfaction measurement on smart speaker intelligent assistants. (Chapter 6)

5. Introducing intent sensitive word embeddings, which can be used as word representation input of natural language processing or information retrieval models. (Chapter 6)

1.2.2 Algorithmic contributions

6. A behavioral user model incorporating both users online digital interaction behavior with a search engine and onsite physical interaction behavior with smart devices in an environment for POI recommendation in smart environment. (Chapter 2)

7. A deep neural behavioral user model for one-shot POI recommendation in a smart environment. (Chapter 2)

8. Test collection augmentation approach to update test collections with an aim of maintaining their reusability. (Chapter 4)

9. User satisfaction modeling by representing a user's interactions with a smart speaker as a sequence of user intents and thus using this sequence to identify user satisfaction. (Chapter 6)

10. User satisfaction modeling by developing intent sensitive word embeddings and then using sequences of these intent sensitive query representations to measure user satisfaction. (Chapter 6)

1.2.3 Empirical contributions

11. (a) A dataset for POI recommendation task in a smart environment which includes real users onsite physical interactions with smart devices in a smart museum and online digital interactions with the museum search engine. (b) Analysis on similarity of users online digital interaction behavior to onsite physical interaction behavior. (c) Analysis of the effect of given seen POIs set-size in the unseen POI recommendation performance (Chapter 2)

12. (a) The TREC contextual suggestion web corpus, consisting of a web crawl of the TREC contextual suggestion collection. (b) A dataset consists of endorsements (end user tags) of the attractions given by the person issuing the request as part of her profile in the TREC contextual suggestion track. (Chapter 3)

13. (a) Analysis of the reusability of the TREC Contextual suggestion test collections (i.e., ClueWeb12 and OpenWeb test collections). (b) Expansion of the TREC Contextual suggestion test collection, which fares much better on the stabler measures and can be used for the evaluation of runs not contributing to the original pools. (Chapter 4)

14. (a) Measuring task and session boundary cut-offs in IA systems. (b) Analysis of the impact of learning phase and domain on task and session length and their cut-off estimation. (Chapter 5)

15. (a) Statistically significant improvements over several baselines in terms of common classification evaluation metrics using our proposed user satisfaction models based on the intent-sensitive query representations. (b) A dataset for user satisfaction prediction and evaluating the performance of different user satisfaction prediction models in IAs. (c) Extensive analyses to assess user satisfaction prediction models in different task types. (Chapter 6)

1.3 Thesis Overview

In Chapter 2, we study how to blend users online interaction behaviors with users onsite interaction behaviors to train a user behavioral model for onsite POI recommendation in smart environments to improve user experience in a smart museum; in Chapter 3, we report our TREC 2016 contextual suggestion track organization effort to create a reusable test collection for contextual POI recommendation problem in a smart city context; and in Chapter 4, we detail how to update contextual suggestion test collection to maintain and improve the test collection reusability.

To improve user experience in performing tasks using smart speaker IAs in their smart homes, in Chapter 5, we study impacts of contextual factors such as learning phase on user interaction behavior to effectively identify tasks and sessions in smart

speaker IAs. Chapter 6 details how intent of users' utterance can be used as a signal of user (dis)satisfaction and how we use intent-sensitive query representation for user behavioral modeling on smart speakers to predict user satisfaction.

Finally, in Chapter 7, we conclude the thesis and discuss limitations and future directions.

Although chapters of the thesis can be read independently, there is a dependency between Chapter 5 and 6 as tasks identified in Chapter 5 is an input of the user satisfaction prediction model in Chapter 6. Furthermore, a part of Chapter 4, which is analyzing reusability of the TREC 2014 contextual suggestion test collection, is one of the motivations of Chapter 3, in which we create a reusable test collection for evaluation of personalized contextual suggestion systems.

1.4 Origins

In this section, we list the publications each chapter is based on and explain the role of each author.

- **Chapter 2** is based on the following papers:
  - S. H. Hashemi and J. Kamps. Exploiting behavioral user models for point of interest recommendation in smart museums. *New Review of Hypermedia and Multimedia*, 24(3):228–261, 2018 [71]
  - S. H. Hashemi and J. Kamps. Where to go next?: Exploiting behavioral user models in smart environments. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 50–58. ACM, 2017 [68]
  - S. H. Hashemi and J. Kamps. Skip or stay: Users behavior in dealing with onsite information interaction crowd-bias. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, pages 389–392, 2017 [70]
  - S. H. Hashemi, W. Hupperetz, J. Kamps, and M. van der Vaart. Effects of position and time bias on understanding onsite users' behavior. In *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval*, CHIIR '16, pages 277–280. ACM, 2016 [77]

  SHH designed the algorithm, ran the experiments, and did most of the writing; JK contributed to the writing; WH and MV helped in collecting experimental data.

- **Chapter 3** is based on the following paper:
  - S. H. Hashemi, C. L. A. Clarke, J. Kamps, J. Kiseleva, and E. M. Voorhees. Overview of the TREC 2016 contextual suggestion track. In *Proceeding of Text REtrieval Conference (TREC)*, 2016 [76]

  SHH designed the algorithm, ran the experiments, and did most of the writing; JKa helped with the algorithm design and contributed in writing; JKa, CC, JKi, EV contributed in designing the experiments.

- **Chapter 4** is based on the following papers:
  - S. H. Hashemi, C. L. Clarke, A. Dean-Hall, J. Kamps, and J. Kiseleva. An analysis of test collection building in dynamic domains. *Under Submission*, 2020 [82]
  - S. H. Hashemi, C. L. Clarke, A. Dean-Hall, J. Kamps, and J. Kiseleva. An easter egg hunting approach to test collection building in dynamic domains. In *NTCIR-EVIA*, pages 1–8, 2016 [74]
  - S. H. Hashemi, C. L. Clarke, A. Dean-Hall, J. Kamps, and J. Kiseleva. On the reusability of open test collections. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 827–830, 2015 [72]

  SHH designed the algorithm, designed and ran the experiments, and did most of the writing; JKa helped with the algorithm design and contributed in writing; AD helped with running the experiments; JKa, CC, AD, JKi, contributed in designing the experiments.

- **Chapter 5** is based on the following paper:
  - S. H. Hashemi, K. Williams, A. El Kholy, I. Zitouni, and P. Crook. Impact of domain and user's learning phase on task and session identification in smart speaker intelligent assistants. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1193–1202, 2018 [80]

  This work was done during an internship at Microsoft in 2017. The task was proposed by IZ; SHH designed the algorithms, ran experiments, and did most of the writing; KW, AK helped with the algorithms design and running experiments. KW, AK, IZ, and PC contributed to the writing.

- **Chapter 6** is based on the following paper:
  - S. H. Hashemi, K. Williams, A. El Kholy, I. Zitouni, and P. Crook. Measuring user satisfaction on smart speaker intelligent assistants using intent sensitive query embeddings. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1183–1192, 2018 [81]

  This work was done during an internship at Microsoft in 2017. The task was proposed by IZ; SHH designed the algorithms, ran experiments, and did most of the writing; KW, AK helped with the algorithms design and running experiments. KW, AK, IZ, and PC contributed to the writing.

The thesis also indirectly builds on the following papers (listed in reverse chronological order):

- M. van Zeelt, F. den Hengst, and S. H. Hashemi. Collecting high-quality dialogue user satisfaction ratings with third-party annotators. In *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval*, pages 363–367, 2020 [158]

- K. Williams, S. H. Hashemi, and I. Zitouni. Automatic task completion flows from web APIs. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1009–1012, 2019 [170]

- S. H. Hashemi and J. Kamps. On the reusability of personalized test collections. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 185–189, 2017 [69]

- S. H. Hashemi, J. Kamps, and W. Hupperetz. Busy versus empty museums: Effects of visitors crowd on users behaviors in smart museums. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 333–334, 2017 [79]

- S. H. Hashemi, J. Kamps, and N. O. Amer. Neural endorsement based contextual suggestion. In *Proceeding of Text REtrieval Conference (TREC)*, 2016 [78]

- S. H. Hashemi, M. Dehghani, and J. Kamps. Parsimonious user and group profiling in venue recommendation. In *Proceeding of Text REtrieval Conference (TREC)*, 2015 [73]

- S. H. Hashemi and J. Kamps. Venue recommendation and web search based on anchor text. In *Proceeding of Text REtrieval Conference (TREC)*, 2014 [67]

2

Exploiting Behavioral User Models for Point of Interest Recommendation in Smart Museums

In this chapter, to improve user experience in smart museums, we focus on modeling both users onsite physical information interaction behavior with smart devices in a smart environment and their online digital information interaction behavior with a search engine to personalize users experience and effectively predict Point Of Interests (POIs) in the smart environment. Our aim is to answer *RQ1: How to model users' information interaction behavior with IoT having an aim of providing a personalized onsite POI recommendation?*

2.1 Introduction

The last decade witnessed a surge of interest in the implementation of Internet of Things (IoT) in different applications, such as smart shopping malls and smart museums, which provide the infrastructure for understanding users' physical interaction behavior and consequently their preferences in interacting with smart environments [12, 14, 57, 70, 77, 89, 136]. This prompts a range of questions: In what ways can tracking people in their real-life behavior and trying to understanding their interaction behaviors be helpful? Is it possible to give effective recommendations to users by tracking them using IoT but without getting any explicit information, like ratings, about their preferences?

Imagine you are at a huge museum like the Louvre in Paris and you want to explore the museum. Usually, it is impossible to visit every single object in a large museums like the Louvre in one day. Furthermore, freely roaming through the museum is more desirable in comparison to the traditional fixed walking route designed in a non-personalized way. Providing personalized experiences for users is highly valuable in this context and will help them to visit all the interesting objects of the museum according to the user's preferences. In this case, how amazing would it be if a contextual recommender system can tell you accurately what to visit without relying on extensive history or explicit feedback from you?

The emergence of applications like the above leads to interest in logging users' onsite

Figure 2.1: Interactive POIs in a museum physical space, consisting of a series of pedestals with screens and actuators integrated into the Roman department of the Allard Pierson Museum of Archaeology in Amsterdam, The Netherlands.

physical information interactions, creating a new and potentially exponentially growing data about physical interaction that resembles current online search engine interaction logs. Although understanding users' search behavior and their information needs based on query logs is well-studied [32, 80, 81, 163], to the best of our knowledge, there has not yet been any study on how to understand users' behaviors and their information needs based on similarities between users' onsite physical and online digital information interaction behaviors. The main contribution of this chapter is to address this research problem by learning a behavioral user model using both onsite physical and online digital user behaviors.

To this aim, users' onsite physical interactions of visits in a museum and users' online query logs of a search engine on the same collection are logged. Onsite physical information interactions are based on unlocking contents of an installed iPad screen at each POI using RFID tags. For privacy reasons we don't have shared IDs, hence users in both sets are un-connected, and we study the typical cold start case where we have no prior history on a visitor to the smart exhibition in the museum yet we have historical data of users' online interactions with the museum search engine. We study how we can use similarity of users' online and onsite information interaction behaviors with an aim of improving onsite POI recommendation at the smart museum. Figure 2.1 shows an example of the museum space with the mentioned installations. In this way, we log users' interactions with POIs and track users' visits in the museum. Figure 2.2 shows the floorplan of an exhibition in a smart museum with an integrated IoT. As it is shown in Figure 2.2, users behave differently after visiting a set of POIs. The walk-through graph of three real users after checking in at POI_1 and POI_2 is plotted. The blue and red paths show walk-through behaviors of two users tend to check-in at POIs one after the other but with different preferences. The green path shows a user who behaves completely different from the other two and does not check-in at POIs one after the other. This figure shows an example of how different users exhibit different

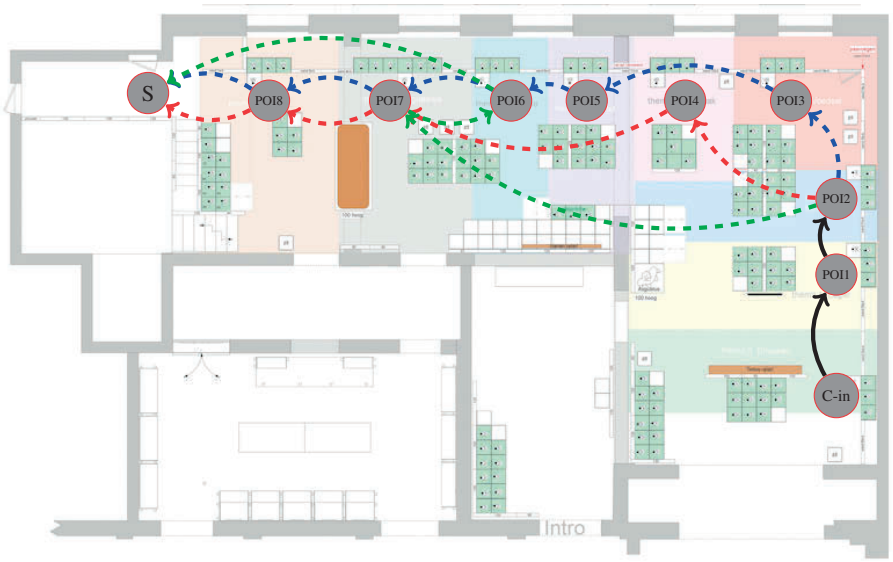


Figure 2.2: Variance in onsite users' behavior after visiting a set of POIs in a museum exhibition shown in Figure 2.1. The figure indicates variance of three visitors' preferences in visiting POIs. Each of them shown by a different color, and the black edges are the ones walked by all the three visitors. $C\text{-}in$ is the check-in station and the S is the check-out station.

onsite physical behavior, which indicates that understanding and prediction of users' onsite physical behaviors can be challenging and difficult.

Understanding users' onsite physical behavior is also challenging as there are external factors in the environment having impact on users' behavior. As it is studied in [77], users' walk-through behavior and their dwell-time interacting with a POI in an exhibition is affected by the position of the POI in the exhibition. They have also observed a decrease in users interests in interacting with technology at the end of an exhibition compared to the start of the exhibition. These external factors lead to position and temporal rank bias in the collected onsite sensor logs [77]. Furthermore, users' behavior is also affected by other visitors around them, which leads to an observation of crowd-bias in collected onsite interaction logs [70].

Such external factors bring an additional complexity to understand users' onsite behavior as it makes users' behavior a combination of "pure" content preferences and other factors like the physical constraints. Moreover, there is a difference in how different users will behave in the presence of external factors as those discussed above. Therefore, understanding users' onsite behavior and preferences in order to provide an effective personalized service in a smart environment is an interesting yet challenging problem. Understanding users' onsite behavior and providing effective personalized POI recommendation become even more challenging in smart museums as in the early stage of launching a smart museum, we do not have access to considerable amount of onsite walk-through sensor logs. Thus, taking advantage of other user preferences signals available for a same collection could be very helpful. To this aim, we study similarity of users' online and onsite preferences by using users' online interaction

behavior signals to model their onsite interaction behaviors. Specifically, we build a graph, in which graph nodes are the POIs available in a smart museum and graph edges are created based on users' click-through behavior on an online search engine providing access to the same museum collection. We then define behavioral features based on the built graph, which are used to create our proposed behavioral user models.

In this chapter, our main aim is to study the **RQ1**: *How to model users' information interaction behavior with IoT having an aim of providing a personalized onsite POI recommendation?* Specifically, we answer the following research questions:

1. *How to understand users' onsite physical behavior and create a behavioral user model that is able to effectively predict relevant unseen POIs?*

2. *How strong are different users' interaction behaviors with IoT in understanding users' preferences?*

    2.1. *Are online digital behaviors similar to onsite physical behaviors? Does understanding online digital users' information interaction behaviors have a positive effect in learning a model to predict unseen relevant POIs and complete users' personalized onsite visits?*

    2.2. *What are the relative importance of each feature extracted based on different users' interaction behaviors in effectiveness of POI recommendation systems?*

3. *How effective is behavioral POI recommendation system in one-shot POI recommendation problem?*

4. *What is the effect of given seen POIs set-size in the unseen POI recommendation performance?*

This chapter builds on and extends the work reported in [68] by providing more detail and explanations of the approach and it's relation to related works, and further analysis such as a study of the impact of number of seen POIs on the performance of the unseen POI recommendation system. The rest of the chapter is organized as follows. In Section 2.2, we review related work on recommender systems and their use in the museum domain, as well as on tracking behavior in smart environments. Our proposed onsite POI recommendation approach is detailed in Section 2.3. The experimental setup and results are discussed in Section 2.4 and 2.5. In Section 2.6, we discuss potential future directions of our study in this chapter. Finally, we present the conclusions and future work in Section 2.7.

2.2 Related Work

In this section, we discuss related work on context-aware recommendation systems, POI recommendation systems, recommendation systems in museums, and the Internet of Things (IoT).

2.2.1 Context-Aware Recommendation Systems

Traditionally, recommender systems deal with applications having just two types of entities, users and items. However, creation of more complex and realistic applications leads to interest in a new line of research about how to incorporate contextual information as an extra dimension into the recommendation systems [76]. There are three ways of incorporating context in the recommender systems: contextual pre-filtering, contextual post-filtering, and contextual modeling [1]. As the later approach is closer to our study in this chapter, we will discuss some of the related research in the contextual modeling.

In order to contextually model the context aware recommendation system, Karatzoglou et al. [101] proposed a multiverse recommendation method based on tensor factorization, which integrates contextual information by modeling data as a User-Item-Context N-dimensional tensor instead of a traditional 2-dimensional User-Item matrix. One problem of this method is the data sparseness, which is proportional to the number of defined contexts in their method. Liu and Aberer [115] proposed to partition the User-Item matrix by grouping ratings of similar context, which could be helpful to decrease the data sparseness. The other problem of the multiverse recommendation method is that it only works for categorical features. To overcome this problem, Rendle et al. [142] proposed to use factorization machines to model contextual information. The above studies are done to model contextual information, however none of them are scalable enough to be effective for the recent exponentially growing data.

2.2.2 POI Recommendation Systems

There have also been many studies to solve the POI recommendation problem in both academia and industry [65, 187]. They generally try to adapt traditional recommendation algorithms to the POI recommendation problem. One line of research includes collaborative filtering and matrix factorization approaches in location-based social networks (LBSNs). Berjani and Strufe [19] proposed regularized matrix factorization, in which they apply personalized collaborative filtering on dimensionally reduced user-POI matrices to minimize the squared regularized error. In addition to the geographical aspects, there is research on POI recommendation that in addition to the geographical dimension also includes the temporal dimension in the matrix factorization framework [59, 63].

Within the POI recommendation literature, there are some studies that are related to ours in the sense that they studied users' check-in behavior [134, 144, 171, 172, 175, 181–183, 185]. As three interesting examples of these related works, Zheng et al. [182] proposed collaborative location activity filtering. Particularly, they used collective factorization to recommend locations or activities to users. To this aim, they used comments having GPS data in a web-based GPS management system as a data source. Moreover, Ye et al. [172] proposed a collaborative POI recommendation algorithm based on geographical influence. To this aim, they used users check-in activities in LBSNs. At last, Scholz et al. [144] studied talk attendance prediction in an academic conference using a link prediction approach. To this aim, they logged talk attendance behavior using RFID tags. However, none of the above studies used both the actual users'

onsite physical information interaction behaviors and users' online digital click-through behaviors.

2.2.3 Recommendation Systems in Cultural Heritage

Another line of related work is research on recommender systems for museum visitors. In museums, although using mobile tour guides cause negative social effects such as less interaction with visitors' fellow group members in a group visit, visitors are interested in using location-aware mobile tour guides, in which they could get information from the guide and spend more time in exhibitions [112]. As many museums have extensive collections of objects which makes it impossible to visit all of them in a single day, requiring visitors to be selective. Thus, personalization become one of the key topics of research in cultural heritage domain [9].

Grieser et al. [62] studied next exhibition recommendation problem in the museum space using visitors history. They applied Naive Bayes learning model using textual description, geospatial proximity and popularity of exhibitions. In their study, popularity baseline, which is one of our defined baseline in this chapter, was reported as the most successful next exhibition recommendation model.

Bohnert et al. [20] studied unseen exhibition recommendation using nearest-neighbor content-based filtering approach by taking visitors explicit ratings of exhibitions as inputs. They did the study using 41 museum visitors as participants. Moreover, Bartolini et al. [15] study recommendation of diverse multimedia data across several web repositories, and arrangement of them in visiting paths. They consider location, number of persons and weather condition as context in their contextual pre-filtering system, and they did the study based on 90 users as participants.

Apart from different recommendation methods being used in the above studies in the museum domain, they are limited in term of number of participants in the experiments. In addition, none of them log and study users' onsite physical information interactions behaviors. In this chapter, we log more than 21,000 users' visits of a museum in a 5 months period in operational practice, and our proposed model is based on users' both online digital and onsite physical information interaction behaviors.

In visiting a museum, recommendations can sometimes be very binary, which leads to either a satisfactory visit or a dis-satisfactory one. For example, a visitor might be in a situation of deciding a path to take from two available ones. The problem of deciding which path to target to take in museums has been addressed in [164] by splitting screen of their mobile tour guide to two parts in order to show both paths and what objects are in their way in each path. This is a critical problem that the authors address by giving information to users to decide themselves. In this chapter, we address this problem by a one-shot POI recommendation system using a deep multilayer perceptron.

Closest in spirit to our work is [70], in which users' onsite physical behaviors in the existence of a crowd of users have been studied. They studied skip or stay behavior prediction in checking in different POIs as a classification problem. Their study is different from ours as they do not investigate on similarities between users' physical and digital behaviors. Furthermore, we study a POI ranking problem in this chapter but they did research on onsite physical interaction behavior classification problem.

2.2.4 Internet of Things

The Internet of Things (IoT) is a network of connected physical objects, in which sensors and actuators are seamlessly embedded in physical environments, and information is shared across platforms to develop a common operating picture [64]. The IoT was first introduced by Kevin Ashton in 1999 in supply chain management context [11]. Then, in the past decade, IoT applied to many applications such as health care systems [28], smart cities [179] and smart museums [68].

Integration of IoT in physical environments provides not only the possibility to collect information from the environment (i.e., sensing) and interact with the environment via actuation, command and control [64], but also the opportunity to use the collected information to provide services to users such as analytics [153] and personalization [52, 68].

As the most relevant line of research to our study in this chapter, Evangelatos et al. present a framework for creating personalized smart environments using wireless sensor networks. Similar to our proposed behavioral user model, their proposed framework can take personalized action based on some predefined profiles including information such as users' age. However, our proposed personalization model is very different from their model as we model users behavior based on their implicit interaction signals collected using sensor logs and personalize a user experience based on the user's behavior. Furthermore, their experimental results is based on just 8 users, which is much lower than the number of users in our experiments based on an operational IoT museum environment. In fact, our experimental results is based on thousands of users' onsite and online information interactions logs.

2.3 POI Recommendation Using Users' Behaviors

This section studies how to predict relevant POIs to the given user and context based on users' interaction behaviors, aiming to answer our first research question: *How to understand users' onsite physical behavior and create a behavioral user model that is able to effectively predict relevant unseen POIs?* To this aim, we first present how the smart museum and our collected user interaction logs look like. Then, after formally stating the POI recommendation problem, we detail our proposed behavioral user models and features extracted for training the model.

2.3.1 POI Recommendation in Smart Museums

There is a growing interests in integration of IoT in museums aiming to provide smart services for museum visitors [6, 10, 30, 31, 61, 121, 140, 150]. In this study, we focus on a specific type of smart museums that aims to understanding users' information interaction behavior based on collected onsite sensor and online click-through interaction logs. In particular, we define a smart museum as:

- **Smart museum** is a museum with exhibitions that are richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in museum visits, and connected through a continuous network.

Figure 2.3: An interactive POI in a museum physical space and a RFID tag as a key.

The data used in this chapter is based on the smart exhibition that is part of the Roman department of the Allard Pierson Museum in Amsterdam, the Netherlands. We aim at modeling users' onsite physical interaction behavior in a smart museum by training a behavioral user model based on a collected sensors' information interaction logs. To this aim, in our smart exhibition RFID tags are provided as a key to access some additional information about objects being shown in the museum. Figure 2.3 shows an example of how these keys are being used to unlock content at each POI. These keys are given to users at the start of the exhibition.

At the start of the museum exhibition, there is a check-in station, at which users can enter their preferences in order to personalize the content being shown in all of the POIs. These preferences are perspectives of the narratives (i.e., Rome, Egypt and Lowlands), language (i.e., English and Dutch), and the user's age range (i.e., Adults and Children). Figure 2.4 shows statistics of a sample of the smart museum visitors' preferences collected at the check-in station. In this sample, we exclude any user session that has missing value for any of the three collected preferences. As it is shown in Figure 2.4, visitors are interested in all available content perspective prepared for POIs. Furthermore, as the smart museum is in Netherlands and it is expected, visitors usually preferred Dutch over English content. Moreover, the smart museum is an archaeological museum and our collected onsite interaction logs indicates that we have more adults visitors compared to children visitors.

After checking in, users are free to put their tags on RFID readers of some or all POIs to unlock contents being shown about objects at the POIs. We are mainly interested in the choice, and order, of POIs visitors choose to interact with. Each POI contains three different archeological objects. Users are free to interact with POIs in any order. They can watch short movies, interact with 3D photos of POIs' objects, or read contents about objects being shown at POIs. At each POI, users are able to change the perspective of narratives and learn about objects from different perspectives. However, their visit will still be personalized based on their preference at the check-in station, and they will see narratives based on their initial choice at the next POI. At last, users might check out in a summary station, in which they might leave their name, gender, birth date and email. By leaving their email, users shows their interests to receive more content about the exhibition in a post-visit scenario.

In addition to the users' onsite physical information interaction logs, we have also

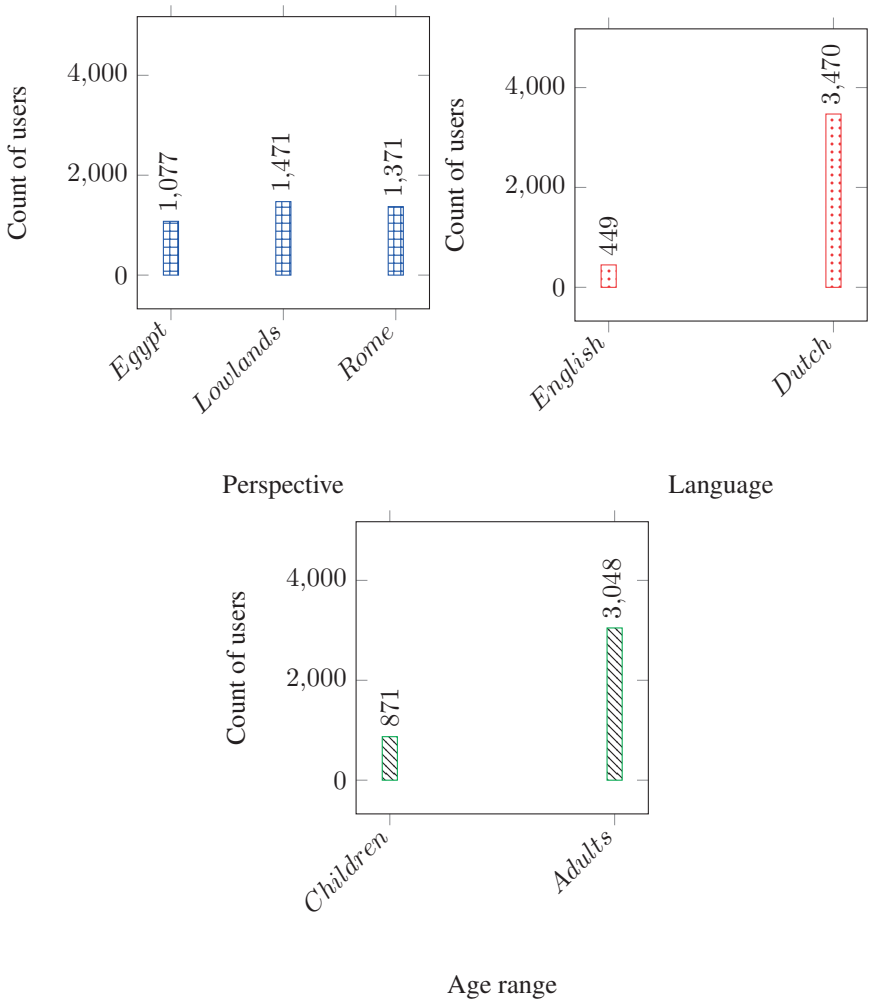


Figure 2.4: Distribution of onsite explicit context chosen by visitors at the check-in station.

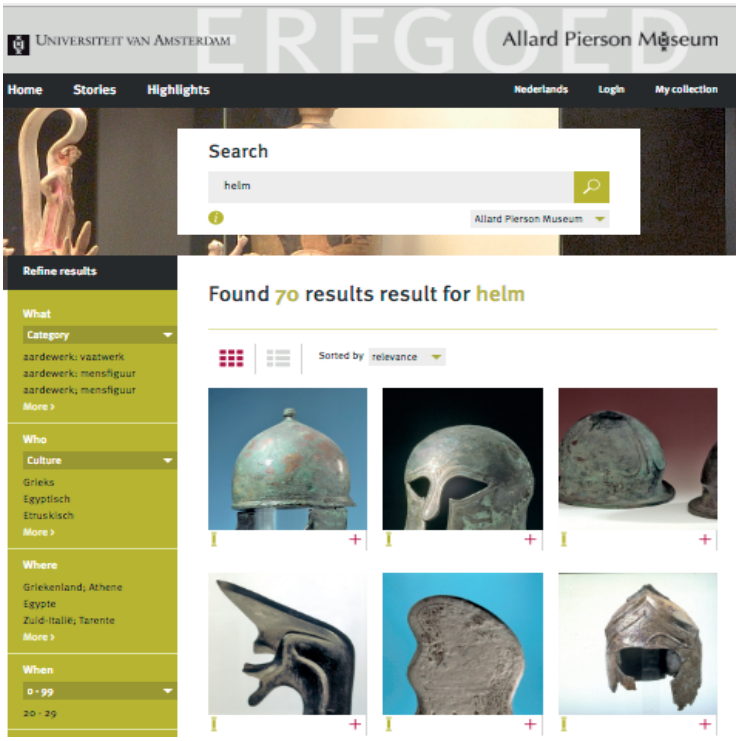


Figure 2.5: A museum's online collection search engine result page.

collected query and click-through logs of the museum search engine. Specifically, when users are in the museum website and explore the museum collection, they might search for an object by issuing a query and then clicking objects being shown in search engine result page (SERP). They might even not issue a query and just click on objects recommended by the museum recommender system. By clicking on objects ranked in the SERP or recommended in the museum search engine first page without issuing a query, users land on the object page, which is shown in Figure 2.5. In the object page, the museum recommender system recommends the most similar objects to the clicked object, which easily lead to click chaining in session. In addition, users might return to SERP and click on another object. They might also revise their query and click on objects retrieved for the given revised query. All these online users' interaction behaviors leads to click chaining that is the basis of our defined online features, which are detailed in Table 2.1.

There are other types of the museum search engine sessions which are not useful for collecting our online features. As all of our online features are based on users' online click-through behavior, we exclude sessions with no click in our data pre-processing. Furthermore, we filter out bot sessions in the data pre-processing.

In smart museums, there are many external factors that might have impact on users' preferences in visiting POIs. For example, a user might be interested in POIs having most popular objects in the exhibition. Furthermore, a user's check-in behavior might

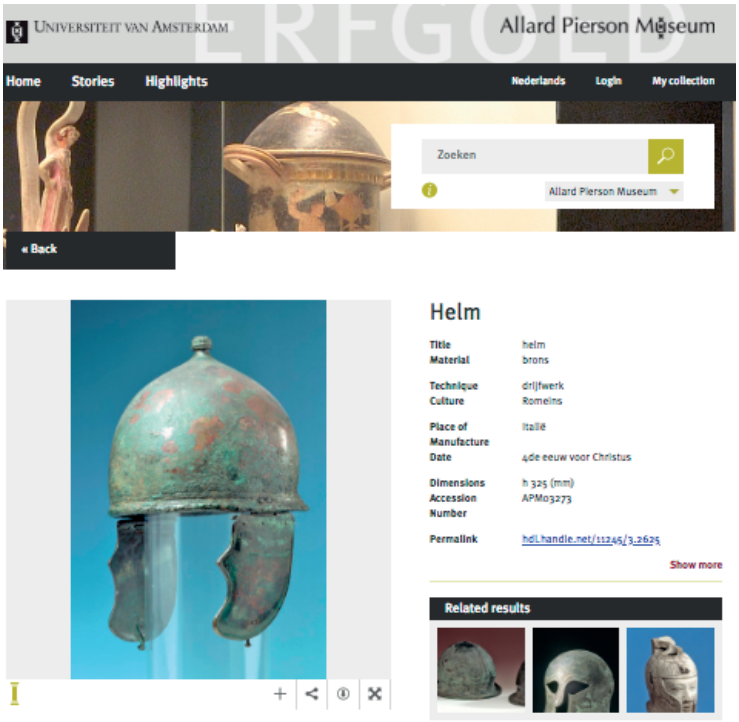


Figure 2.6: A museum's online collection search engine landing object page including related objects recommended to users based on clicking on an object presented in the search engine result page.

be affected by location of POIs presented in the museum [77] or even visitors' crowd in the museum [70]. In addition to all these external factors, users' preference play a major role in their choice to visit an unseen POI after visiting a set of POIs. Users' behavioral dynamics, due to existence of all these factors, makes it very challenging to predict users' next check-in interaction after visiting a set of POIs. To address this problem, in addition to explicit context given by users at the start of an exhibition, we try to implicitly capture context by user's choice of visiting a set of POIs in the physical environment. In the rest of this section, we first state unseen POI recommendation problem based on a set of seen objects in a smart museum, and then we detail our proposed model to address this problem.

2.3.2 Problem Statement

Let $\mathbf{u} = \{u_1, u_2, ..., u_i\} \subset U^i$ be a subset of users visited a smart environment, $\mathbf{c}_{seen} = \{c_1, c_2, ..., c_j\} \subset C^j{}_{seen}$ a subset of seen or occurred contexts, and $\mathbf{p}_{seen} = \{p_1, p_2, ..., p_k\} \subset P^k{}_{seen}$ a subset of seen POIs. Then, let $\mathbf{R}_{seen} \in \mathbb{R}^{i \times j \times k}_{seen}$ be a user-context-POI matrix containing i users, j seen contexts and k seen POIs. Value $r_{i,j,k} \in \mathbf{R}_{seen}$ refers to the visit frequency of user i, in context j to the POI k. In this chapter, due to the fact that museum visitors rarely check in to a POI more than once, we have used binary seen or unseen values rather than considering the frequency.

Having above information about users, given a subset of unseen contexts (i.e., $\mathbf{c}_{unseen} = \{c_1, c_2, ..., c_m\} \subset C^m{}_{unseen}$), and a subset of unseen POIs (i.e., $\mathbf{p}_{unseen} = \{p_1, p_2, ..., p_n\} \subset P^n{}_{unseen}$), the behavioral unseen POI recommendation problem is estimation of $r_{i,m,n} \in \mathbf{R}_{unseen}$ based on users interaction behaviors with the seen POIs, in which $\mathbf{R}_{unseen} \in \mathbb{R}^{i \times m \times n}_{unseen}$ is a user-context-POI matrix containing i users, m unseen contexts and n unseen POIs.

In order to model the set-based contextual POI recommendation, we cast the context-aware recommendation problem to a binary classification problem, in which relevant POIs are labeled 1 and irrelevant ones labeled 0. In this way, we try to learn a behavioral model to predict relevant unseen POIs to the given user and context based on the user's interaction behaviors in the context. Then, relevance probability of POIs to the user and context pairs will be used to rank the unseen POIs. To this aim, a set of features that represent users' interaction behaviors in given contexts is defined.

2.3.3 Feature Set

In order to learn an effective model to rank POIs, we have extracted 18 different features. As shown in Table 2.1, we have classified features to three sets, namely, explicit context, onsite and online.

The *explicit context features* refer to information explicitly given by users about the context. In our study, we collected users' gender, their preferred language, their age range and their chosen perspective of the narratives at the exhibition. Previous study on these explicit contexts [77] shows that users behave differently in these different contexts. For example, as it is discussed in [77], children tend to spend less time in front of the POI about death. Therefore, it seems a reasonable set of features to consider as explicit contexts. Furthermore, the content being shown in the exhibition at each POI

Table 2.1: Defined features to predict relevant unseen POIs to users after visiting a set of POIs

Feature	Category	Description
f_1	Explicit Context	Gender (e.g., Female)
f_2	Explicit Context	Language (e.g., English)
f_3	Explicit Context	Visitor age range (e.g., Adults)
f_4	Explicit Context	Chosen perspective (e.g., Roman)
f_5	Onsite	Seen POIs set size.
f_6	Onsite	Content-based relevance score of a POI candidate to a profile created using seen POIs' content that was shown onsite
f_7	Onsite	Unseen POI's PageRank in onsite visits walk-through weighted graph built based on a train set.
f_8	Onsite	Unseen POI's PageRank in onsite visits walk-through unweighted graph built based on a train set
f_9	Onsite	Unseen POI's centrality in onsite visits walk-through graph built based on a train set.
f_{10}	Onsite	Minimum distance of the seen set of POIs to the POI candidate in the onsite visits walk-through graph built based on a train set
f_{11}	Onsite	Median distance of the seen set of POIs to the POI candidate in the onsite visits walk-through graph built based on a train set
f_{12}	Onsite	Mean distance of the seen set of POIs to the POI candidate in the onsite visits walk-through graph built based on a train set
f_{13}	Online	Unseen POI's PageRank in Online click-through weighted graph built based on a train set
f_{14}	Online	Unseen POI's PageRank in Online click-through unweighted graph built based on a train set
f_{15}	Online	Unseen POI's Centrality in Online click-through graph built based on a train set
f_{16}	Online	Minimum distance of the seen set of POIs to the POI candidate in the Online click-through graph built based on a train set
f_{17}	Online	Median distance of the seen set of POIs to the POI candidate in the Online click-through graph built based on a train set
f_{18}	Online	Mean distance of the seen set of POIs to the POI candidate in the Online click-through graph built based on a train set

is personalized, which implicitly has impact on users onsite interaction behavior.

The second group consists of *onsite features* which are a set of implicit behavioral features collected during the interactions in the smart environment. In particular, we use onsite features extracted based on user walk-through data. Specifically, f_5 is the number of seen POIs, which can be a signal of visitors' expertise in interacting with the POIs. In addition, it can be considered as a confidence indicator of some other features' scores like f_6. Whereas f_6 is the content-based filtering score of POI candidate based on the profile built using the seen POIs. This content-based filtering score is calculated based on the onsite POI descriptions and users' onsite interactions. That is why it is considered as one of the onsite features in our feature classification.

In addition to f_5 and f_6, we build users' walk-through graph using their onsite interactions with POIs based on the train set onsite information interaction logs, and calculate the further f_7, f_8, f_9, f_{10}, f_{11} and f_{12} features. Details of these features are available in Table 2.1. In particular, f_7 is unseen POI's PageRank in the onsite visits walk-through weighted graph. Weight of a link from POI_a to POI_b is the number of times that visitors visited POI_b after checking in at POI_a. The main motivation behind using pagerank rather than link popularity of POIs is the fact that pagerank helps minimizing the effect of position rank bias of the POI1. It is shown in [77] that there is a position rank bias in smart museums and it is more likely that users check in at POI1, which is the closest POI from the check-in station. This leads to high degree of both incoming and outgoing node degree for POI1. Using pagerank give less importance for incoming links from POIs with many outgoing links (e.g., POI1), which minimizes the possible bias on users' behavior based on available external factors. On the other hand, f_9 is centrality feature that can capture popularity in the walk-through graph.

The third group consists of *online features* refers to a set of features based on online interaction logs based on the collection information as offered on the museum's web site. The features are defined in a similar way as we have modeled the onsite selected POIs using the onsite users' interactions logs. However, the feature calculation is entirely based on the prior online click-through graph of the museum search engine. As said before, we assume a cold start scenario, where no mapping between users at the smart exhibition and the online logs, hence no online prior history of the particular visitor. The online click-through graph is filtered to the objects available at onsite POIs. In this study, each onsite POI contains 3 different museum objects. We merge all the objects related to each POI as one node, and the click-through graph's edges are aggregated from all the edges of POIs' objects. As a result, same as onsite walk-through graph, the online click-through graph has onsite POIs as nodes. Details of these features are available in Table 2.1.

2.3.4 Learning Model

In order to learn a set-based behavioral POI recommendation model, we have implemented a logistic regression classifier and a deep neural multilayer perceptron with dropouts to estimate relevance of each POI to the given user after visiting a set of POIs. The logistic regression classifier and the deep multilayer perceptron have been trained separately based on each group of features extracted using different users' information interaction behaviors to study which user information interaction behavior is more

effective in understanding users' preferences in their interactions with the IoT in smart environment. In the rest of this section, we will detail the logistic regression and the deep multilayer perceptron implemented for the set-based behavioral POI recommendation.

Logistic Regression

Logistic regression classifier is a linear classifier that transparently helps understand contribution of each feature in estimation of POIs relevancy. In fact, we would like to know which trained logistic classifier performs better and why. To this aim, we train different logistic regression classifiers based on different feature sets using different users' interaction behaviors.

In order to learn a logistic classifier, we use variable $c \in \{0, 1\}$ to show relevance of a POI to a user in a context. Specifically, $P_\theta(c = 1|u, c, p)$ is the relevance score of the POI p to the user u and the context c, in which θ is unknown parameters learned using maximum likelihood estimation (MLE) based on the train set. Given the relevance judgments r of each POI p_k to a user u_i and context c_j in the train set, the likelihood L of the train set is as follows:

$$L = \prod_{i=1}^{|U|} \prod_{j=1}^{|C|} \prod_{k=1}^{|P_{seen}|} P_\theta(c = 1|u_i, c_j, p_k)^r P_\theta(c = 0|u_i, c_j, p_k)^{1-r},$$

in which we assume relevance judgments r are generated independently. We model $P_\theta(c = 1|u_i, c_j, p_k)$ by logistic function on a linear combination of features created based on each specific group of users' information interaction behaviors. Then, we optimize the unknown parameters θ by maximizing the following log likelihood function:

$$\theta^* = argmax_\theta \sum_{i=1}^{|U|} \sum_{j=1}^{|C|} \sum_{k=1}^{|P_{seen}|} rlogP_\theta\left(c = 1|u_i, c_j, p_k\right) + (1 - r)\, logP_\theta\left(c = 0|u_i, c_j, p_k\right).$$

In order to turn the logistic classifier scores to probabilities, we have used the softmax function:

$$S\left(y_i\right) = \frac{e^{y_i}}{\sum_j e^{y_j}},$$

in which y_i is the logistic classifier score, and $S(y_i)$ is the output relevance probability of our behavioral POI recommendation model. At last, we rank unseen POIs based on the logistic classifier output probability of POIs' relevancy being estimated based on features created using interaction behaviors of a given user in a context.

Deep Neural Multilayer Perceptron

In this subsection, we investigate on a deep neural multilayer perceptron (MLP) by an aim of improving effectiveness of the POI recommendation to be used in critical one-shot POI recommendation applications. The motivation behind the critical one-shot POI recommendation is that an irrelevant recommendation sometimes has a very

negative effect in users' experience in a way that they might be incorrectly guided to an uninteresting department of a museum that leads to a dissatisfied experience. In this model, for each user in a context, our main goal is to recommend a POI which is highly relevant to them. In the one-shot POI recommendation, we do not care about relevant POIs retrieved after rank 1. In the rest of this section, we detail our deep multilayer perceptron with an aim of improving effectiveness of POI recommendation to be used for the critical one-shot POI recommendation problem.

In order to learn a set based behavioral POI recommendation and learn users' onsite complicated physical behaviors, we have used a deep MLP neural network with 3 hidden layers having 326 units. To learn an effective model and overcome overfitting problem, we have used a dropout feedforward neural network. Let $l \in \{1, 2, 3\}$ be the index of the hidden layers of the network. Let $z^{(l)}$ be the vector of input to layer l and $y^{(l)}$ be the vector of outputs from layer l. The dropout neural network is modelled as follows for any hidden unit i and $l \in \{0, 1, 2\}$ [90, 152]:

$$\begin{aligned} r^{(l)} &\sim Bernoulli(p), \\ \tilde{y}^{(l)} &= r^{(l)} * y^{(l)}, \\ z_i^{(l+1)} &= w_i^{(l+1)} \tilde{y}^{(l)} + b_i^{(l+1)}, \\ y_i^{(l+1)} &= f(z_i^{(l+1)}), \end{aligned}$$

where $r^{(l)}$ denotes a vector of independent Bernoulli random variables having probability p of being 1, $\tilde{y}^{(l)}$ is thinned outputs created by multiplying a sample of $r^{(l)}$ vector by outputs of layer l (i.e., $y^{(l)}$) and used as input for the next layer $l + 1$, $w^{(l)}$ and $b^{(l)}$ are weights and biases at layer l, and f is an activation function, which is rectified linear units (ReLUs) in our setup. This process is done at each layer.

Following prior research in neural network domain, we have used $p = 0.5$ in our dropout network. This value is reported as a close to optimal value for a wide range of networks in different applications [152].

In the learning phase, the derivatives of the loss function are backpropagated through the dropout network. The dropout network is trained using the stochastic gradient descent (SGD) algorithm with mini batches, which is widely used algorithm for training neural networks. The learning rates are adjusted based on adaptive gradient algorithm (AdaGrad) [49]. In the test phase, the sub-network is used without dropout, but the weights are scaled as $W_{test}^{(l)} = pW^{(l)}$.

For the classification purpose and having probabilities as outputs, we have used Logistic classifier in the last layer. The logistic classifier in the last layer is trained same as the logistic regression classifier being discussed in previous subsection. The only difference is that, in the logistic classifier being used in the last layer, we model $P_\theta(c = 1|u_i, c_j, p_k)$ by logistic function on a linear combination of inputs from the last hidden layer units' outputs. At last, the final relevance probability of $P_\theta(c = 1|u_i, c_j, p_k)$ is used to rank unseen POIs based on features created using interaction behaviors of a given user in a context.

2.4 Experimental Setup

In this section, we describe our experimental setup. We first describe the data set used in this chapter, and second detail the evaluation methodology used in this study.

2.4.1 Dataset

The dataset of this study is based on onsite physical and online digital interaction logs collected at an archeological museum. Onsite physical interaction logs are collected using sensors available in the museum, and the online digital interaction logs are based on click-through behavior of users.

In this chapter, 5 months onsite physical interaction logs of the museum with more than 21,000 sessions is used, which leads to 3,925 high-quality onsite sessions to be used for evaluation purposes.

The online features, detailed in Table 2.1, have been extracted based on 18,001 high-quality sessions created based on a common time-oriented session identification approach in search engines using 30 minutes inactivity time as session cut-off boundary [51, 145]. The main assumption is that a long period of inactivity between a user's activities indicates the user is probably no longer active, which leads to ending the session.

2.4.2 Evaluation Methodology

In our collected onsite information interaction logs, about 16,000 out of 21,000 sessions either did not have any interactions with POIs or they did not check out at the summary station, and about 1,000 of them had interactions with all the POIs. In order to avoid bias over users who are interested in visiting all or none of the POIs at the museum, we exclude all sessions have checked in at all or none of the POIs at the exhibition. As a result of this preprocessing step, 3,925 out of 21,000 high-quality onsite information interaction sessions remain for creating the test collection.

Considering the walk-through graph, for each user in a session and at each checked-in POI during their visit, we created a test collection using the seen set of POIs, the user and the explicit contexts as the query and the unseen POIs as the candidates, for which we have judgments based on the user's session. Basically, we know which POI candidates are visited by the user and consider them as relevant POIs. The rest of the POIs are considered as irrelevant POIs.

Doing the above procedure in building the test collection leads to create a contextual set-based POI recommendation test collection having 1,083,623 judgments. Table 2.2 shows an example of records created using a user session. To test our proposed model, in order to avoid overfitting, we have done five-fold cross-validation, in which for each fold as a test set, three out of the four remained folds randomly sampled and used as a train set, and the remained fold used as a validation set. We repeat the process for all the five folds and report the average of the evaluation metrics.

Table 2.2: An example of records created for the test collection using a user session. The judgments are based on seen POI set-size 2 and 3

Query context	Seen POI set	Candidate	Relevance
c_1	$\langle \mathrm{POI}_1, \mathrm{POI}_2 \rangle$	POI_3	0
c_1	$\langle \mathrm{POI}_1, \mathrm{POI}_2 \rangle$	POI_4	1
c_1	$\langle \mathrm{POI}_1, \mathrm{POI}_2 \rangle$	POI_5	0
c_1	$\langle \mathrm{POI}_1, \mathrm{POI}_2 \rangle$	POI_6	0
c_1	$\langle \mathrm{POI}_1, \mathrm{POI}_2 \rangle$	POI_7	1
c_1	$\langle \mathrm{POI}_1, \mathrm{POI}_2 \rangle$	POI_8	0
c_1	$\langle \mathrm{POI}_1, \mathrm{POI}_2, \mathrm{POI}_4 \rangle$	POI_3	0
c_1	$\langle \mathrm{POI}_1, \mathrm{POI}_2, \mathrm{POI}_4 \rangle$	POI_5	0
c_1	$\langle \mathrm{POI}_1, \mathrm{POI}_2, \mathrm{POI}_4 \rangle$	POI_6	0
c_1	$\langle \mathrm{POI}_1, \mathrm{POI}_2, \mathrm{POI}_4 \rangle$	POI_7	1
c_1	$\langle \mathrm{POI}_1, \mathrm{POI}_2, \mathrm{POI}_4 \rangle$	POI_8	0

2.4.3 Evaluation Metrics

For the evaluation of the defined set-based behavioral POI recommendation task, we cast the problem to a ranking task and use mean reciprocal-rank (MRR), mean average precision (MAP) and R-precision ($R\text{-}Prec$) as metrics that are effective to evaluate proposed models. Moreover, in order to evaluate the one-shot POI recommendation systems, we use precision at rank 1 ($P@1$) as an evaluation metric.

The MRR is the average of the reciprocal ranks of the first relevant result for a set of queries Q as:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}.$$

In our experiments, Q is a set of 1,083,623 queries (user and context pairs). In MRR, $rank_i$ represents rank of first relevant POI for a given pair of user and context. Precision at rank n (i.e., $p@n$) is used in number of evaluation metrics in this study, which is defined as follows:

$$p@n = \frac{\#\ relevant\ POIs\ in\ top\ n\ results}{n},$$

where n is the rank. For a single query, AP is defined as the average of the $p@n$ values for all relevant POIs as:

$$AP = \frac{\sum_{n=1}^{N} p@n \times rel(n)}{R},$$

in which N is the number of retrieved POIs candidates, and $rel(n)$ is a binary function indicating the relevance of a POI to a given user and context pair at a given rank. A POI is relevant to a user and context pair, if the user checks in at the POI at that visit. MAP is the mean value of the APs computed for all queries. $R\text{-}Prec$ is precision at rank R where R is the number of relevant candidates for the given query. At last, $P@1$ is the precision at rank 1.

2.4.4 Baselines

In this section, we detail the baselines created for the evaluation purposes.

Popularity

The popularity based recommendation ranks POIs candidates according to their popularity scores. According to previous evaluation studies in recommender systems such as [88], systems recommending very popular items can guarantee that users will like most of the recommended items. Moreover, the popularity baseline is usually used in evaluation of personalized recommendation systems and it is informed as a competitive baseline [116].

In this chapter, the popularity is computed as the number of users who checked in at each POI. Therefore, regardless of what POI has been already seen by a user, the popularity baseline recommends the most popular POIs according to other users who checked in at the POIs before.

Bias-Based Filtering

In both physical and digital worlds, external factors has impact on users' behavior with information systems [68, 70, 77]. As a result, assuming existence of the same external factors in the physical smart environments, we could take advantage of them and predict the next POI based on users' status in the environment. Although the bias-based filtering baseline could be hard-to-beat, it would not be a very useful recommender system in practice. Such a baseline is not based on users' interests and their profile. They are just predicting users next move using biases and external factors in the environment.

As Hashemi et al. [77] discussed, there are some biases in onsite user information interaction logs. They introduces the walk-through position-bias that shows users tend to visit POIs one after the other from check-in to check-out stations. They also observed time-rank bias that indicates users tend to spend less time at the end of exhibitions. Considering these two biases, the probability of checking in at a POI is proportional to the distance from the Check-out station. Therefore, in all experiments of this chapter, the bias-based baseline ranks POIs based on their distance from the check-out station.

Content-Based Filtering

As descriptions of POIs in museums are well curated, they are an informative source of information that makes content-based filtering as an effective baseline in this domain. In this study, each POI contains three museum objects with reach descriptions. In order to build a content-based filtering model, we build a profile of each user after visiting a set of POIs using Language Modeling framework. Each profile's language model is based on all seen objects of $\mathbf{p}_{seen}$.

Since we have profiles of users at each context based on their seen POIs, KL-Divergence of each unseen POI's language model and the profile is considered as content-based filtering scores for ranking unseen POIs.

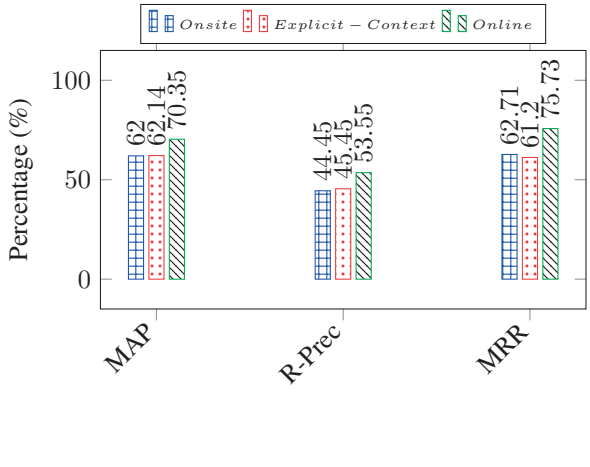


Figure 2.7: Effectiveness of different types of users' interaction behavior in understanding their onsite preferences.

2.5 Experimental Results

In this section, we provide answer to the research questions stated in the introduction section.

2.5.1 POI Recommendation using Users' Information Interaction Behaviors

This section answer our second research question: *How strong are different users' interaction behaviors with IoT in understanding users' preferences?*

To this aim, we have used each of the three groups of features extracted based on each information interaction behaviors to train a POI recommendation system. Specifically, we have trained three different logistic regression classifiers, which are trained based on: 1) the explicit context features (i.e., Logistic Regression-Explicit Context) 2) the onsite features (i.e., Logistic Regression-Onsite) and 3) the online features (i.e., Logistic Regression-Online).

In the rest of this subsection, we first investigate whether users' online digital interaction behaviors are similar to the users' onsite physical behavior. Then, we detail relative importance of each feature extracted based on features' weights being learned by logistic regression classifiers using each type of users' interaction behaviors with an aim of understanding users' behaviors.

Onsite Physical Behavior vs. Online Digital Behavior

We first look at the question: *Are online digital behaviors similar to onsite physical behaviors? Does understanding online digital users' information interaction behaviors have a positive effect in learning a model to predict unseen relevant POIs and complete users' personalized onsite visits?*

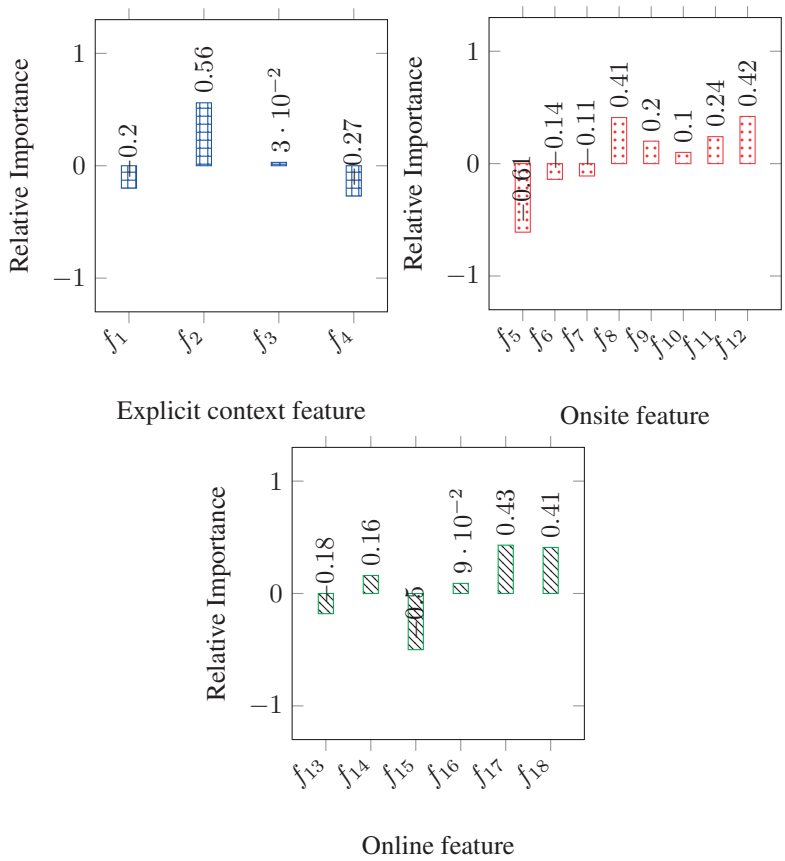


Figure 2.8: Features' relative importance in POI recommendation trained based on each group of users' information interaction behaviors.

In order to answer this research question, we compare POI recommendation systems trained based on each type of interaction behavior. As shown in Figure 2.7, the POI recommendation system trained based on users' online digital interaction behavior is not only as good as the other POI recommendation systems being trained based on either explicit context or onsite interaction behaviors, but also is performing better than them in terms of all common tested information retrieval metrics.

This experiment indicates that availability of the considerable amount of online interaction logs in comparison to onsite interaction logs leads to training an effective onsite POI recommendation system based on users' online digital interaction behaviors. As we achieve an effective onsite POI recommendation system based on users' online digital interaction behaviors, we conclude that there is a similarity between onsite physical and online digital information interaction behaviors.

Table 2.3: Set-based one-shot POI recommendation baselines effectiveness

Run	P@1	MRR
Content-based Filtering	57.45	75.68
Popularity	60.86	77.67
Bias-Based Filtering	**61.57**	**77.71**

Table 2.4: Set-based one-shot POI recommendation effectiveness comparison between the *Deep MLP-Online* and the best baseline. * indicates the improvement is statistically significant ($\rho < 0.05$)

Run	P@1		MRR	
Bias-Based Filtering	61.57		77.71	
Logistic Regression-Online	56.97		75.73	
Deep MLP-Online	**75.81**	(**23.12%***)	**86.39**	(**11.17%***)

Features Relative Importance in Understanding Users' Interaction Behaviors

We now look at the question: *What are the relative importance of each feature extracted based on different users' interaction behaviors in effectiveness of POI recommendation systems?*

To this aim, we normalize features' weights being learned in each logistic regression classifier trained for each group of features separately. Then, average of the normalized features' weights over the 5-fold cross-validation are reported and compared in Figure 2.8.

As it is shown in Figure 2.8, among the explicit context interaction, the chosen language (i.e., f_2) at the start of museum visits is relatively more important in comparison to other explicit context based features. Furthermore, mean distance of the seen POIs to a POI candidate in the onsite visits' walk-through graph (i.e., f_{12}) has relatively more importance in comparison to other onsite interaction behavior based features. Regarding the online interaction behaviors, median distance of the seen set of POIs to the given candidate in the online click-through graph (i.e., f_{17}) is relatively more important than other online features in the effectiveness of the POI recommendation systems.

2.5.2 One-Shot POI Recommendation Using Users' Interaction Behaviors

This section answer our third research question: *How effective is behavioral POI recommendation system in one-shot POI recommendation problem?* To this aim, we first study effectiveness of the discussed baselines in one-shot POI recommendation problem. Table 2.3 shows effectiveness of the baselines in terms of $P@1$ and MRR metrics. Experimental results indicate that the Bias-based filtering baseline performs better than the other baselines in terms of both one-shot POI recommendation evaluation metrics. One possible explanation of this could be that users' interaction behaviors is

highly affected by external factors in physical environments [70, 77], which leads to more predictable user behavior in the existence of those external factors. Thus, the bias-based filtering baseline is even performing slightly better than the popularity baseline, which is a hard-to-beat baseline according to previous studies in recommendation systems in cultural heritage [116].

In order to evaluate effectiveness of our proposed one-shot POI recommendation model, we study effectiveness of the implemented deep multilayer perceptron in one-shot onsite POI recommendation problem in comparison to the best performed baselines as well as the logistic regression POI recommendation system. Table 2.4 shows performance of the best deep multilayer perceptron (i.e., Deep MLP) and logistic regression classifiers, trained based on online digital interaction behaviors, in terms of $P@1$ and MRR.

In this experiment, we just focus on the results based on $P@1$ and MRR as in one-shot POI recommendation problem, we just care about the first ranked unseen recommended object. Thus, $P@1$ is the main metric in evaluation of this problem. In the evaluation of this experiment, we have also used the MRR metric as a representative of the early precision based metrics.

As it is shown in Table 2.4, the deep MLP significantly improves the best competitive baseline (i.e., Bias-Based Filtering) in one-shot POI recommendation. In particular, the deep MLP has 23.12% improvement over the bias-based filtering baseline in terms of $P@1$, which is the metric that measures as closely as possible the one-shot POI recommendation performance. This experimental result shows that our proposed deep MLP one-shot POI recommendation system results in very high precision, suggesting it's practical use to created an enhanced personalized experience for this critical application.

2.5.3 Impact of Seen Set Size

This section answer our research question: *What is the effect of given seen POIs set-size in the unseen POI recommendation performance?*

In this experiment, we analyze impact of different seen POIs set-size in the effectiveness of the final POI recommendations. As it is shown in Figure 2.9, overall performance of the recommendations are improved while users interact more with the POIs and see more POIs. However, there are some biases in the users' onsite information interaction logs that add some noises in the observed patterns based on the seen POIs set-size.

For example, due to the observed position-rank bias in users' onsite behavior, POI_4 has a higher chance of being the fourth seen POIs in users' visits. It seems the POI_4's location in the exhibition is the start location of a strong position-bias in which users tend to visit POIs one after the other, except when there is a crowd of visitors in front of the next POI. This makes it more difficult to understand users' preferences in checking in at unseen POIs. This may explain why there is a slightly decrease in recommendations performance at seen POI set-size 4.

In real applications, we may always have external factors like the topology of POIs in the physical space contributing in users' behavior, which in this case, decreases recommendations performance at four seen POIs. However, in an experimental environment that all external factors and biases are avoided, we can improve recommendations

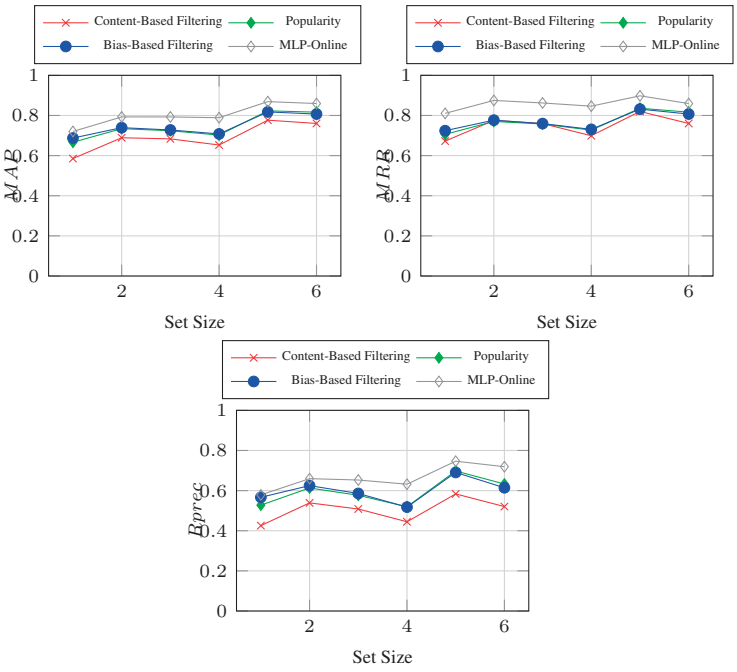


Figure 2.9: Effects of seen POIs set size on the performance of the best proposed model and baselines.

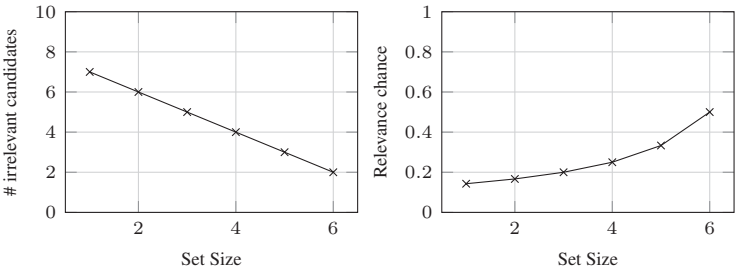


Figure 2.10: Impact of seen POIs set size on one-shot unseen POIs recommendation based on number of irrelevant candidates in contrast to just one relevant POI (left figure) and relevancy chance of a random recommended unseen POI (right figure).

effectiveness by having more seen POIs and creating a richer profiles.

Experimental result shown in Figure 2.9 indicates that the effectiveness of our best proposed POI recommendation model (MLP-Online) is improved by increasing number of seen POIs in sessions. However, the improvement is not just due to obtaining more history about the user profile. Specifically, in one-shot POI recommendation problem, according to the number of available candidates in different seen POIs set-size, the one-shot POI recommendation problem becomes easier when a smaller number of unseen POIs remains, compared to the start of exhibition's visit.

Figure 2.10 shows what is chance of recommending relevant POI in one-shot POI recommendation is by randomly recommending a POI at each seen POIs set-size. Specifically, when seen POIs set-size of a user visit is equal to one, one-shot POI recommendation system has seven different POI candidates to recommend in our experiment. As a result, by just randomly recommending a POI, it would have $1/7 \sim 14\%$ chance of recommending a relevant POI to the user. On the other hand, if a user visited six POIs and has two unseen POIs in their visit, we would have $1/2 \sim 50\%$ chance of recommending a relevant POI to the user by randomly recommending a POI.

As it is discussed in this experiment, our proposed model based on online features is much less affected by the available biases in the users' onsite information interaction logs in comparison to all the other models. This experiment shows that the proposed model is performing better than all the baselines at any seen set-size. In fact, although one-shot POI recommendation problem is relatively more difficult when a user's seen POI set-size is low and have relatively higher number of candidates compared to later stage of their visit, the improvement is even higher in lower seen set-sizes. One possible explanation of this is that as the MLP-online trains the one-shot POI recommendation model based on a larger number of hyper-parameters compared to baselines, it could be able to have a greater improvement over baselines when the problem is harder to address. In the next section, we discuss what would be future directions of our study in POI recommendation in smart environments.

2.6 Future Directions

As shown in this chapter, we have achieved a high performance for next POI recommendation problem using our proposed model. This one step recommendation problem is a key application for museum exhibition navigation, or more generally next step recommendation in smart environments, but there are other interesting applications that suggest themselves. In particular, can we recommend a whole route which may require additional aspects such as considering length or diversity, that are not captured by the one step recommendation problem. In future work, we plan to study the problem of route prediction in smart environments based on seen POIs profile logged by onsite sensors.

Let us discuss an illustrative example. In a sample of the onsite sensor logs of the smart museum being studied in this chapter, we have got 136 visitors who have checked-in POI_1, POI_2 and POI_4 but decided to skip interacting with POI_3. At this point, it would be interesting to recommend a personalized route to users. According to our observation, users behave differently in checking-in the remained unseen POIs,

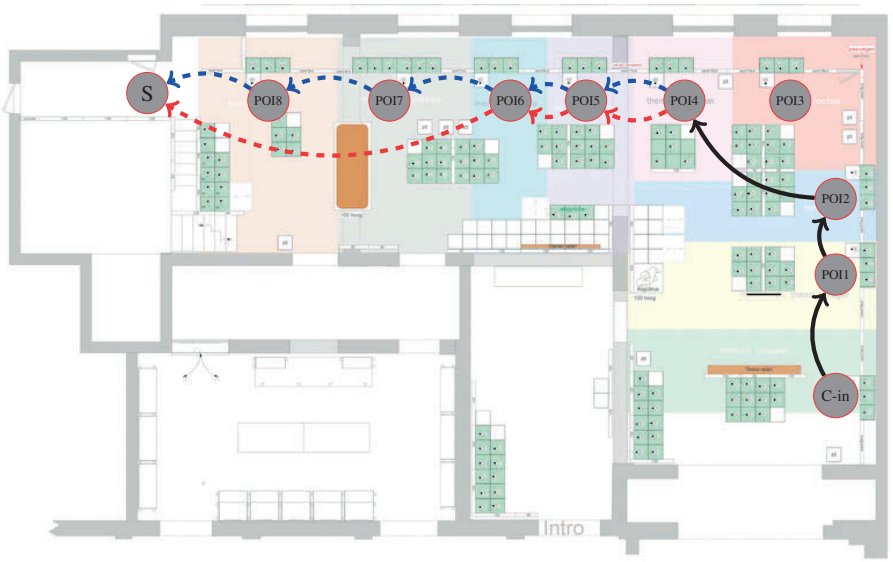


Figure 2.11: Users' unseen POI routes after visiting a set of POIs, namely, POI1, POI2, and POI4 by skipping the POI3. The figure demonstrates two most popular unseen POI routes based on a real traffic in a smart museum. Each of them shown by a different color, and the black edges are the ones walked by all the three visitors. $C\text{-}in$ is the check-in station and the S is the check-out station.

namely, POI_5, POI_6, POI_7, and POI_8. In particular, 18% (24 out of 136) of the sampled users chose to visit all the remained POIs one after the other (blue dashed lines in Figure 2.11), which is the most popular route. The second popular route is checking-in POI_5 and POI_6 but skip interacting with POI_7 and POI_8 (red dashed lines in Figure 2.11), which was based on 12 % (16 out of 136) of the sampled users' interactions.

As it is shown above, visitors have different preferences in checking-in different POIs. Thus, understanding users' onsite interaction behavior and recommending the best route to take in smart environments is a challenging problem to study. We do not discuss ideas on how to model users' behavior to predict unseen objects' route, however, we have observed different behaviors based on some explicit preferences that were given by visitors. For example, among the 24 visitors who decided to check-in all the remained POIs of the above example, 10 out of 24 were interested in narratives from "low lands" perspective in contrast to 7 out of 24 who where interested in narratives from "Rome" perspective. The rest were interested in narratives from "Egypt" perspective. As we have observed for the POI recommendation in smart environments problem, using explicit-context, onsite and online features leads to effective POI recommendation models. Thus, in the future work, using the mentioned features might be also a reasonable features to start for the unseen route recommendation problem in smart environments.

As it is discussed in previous section, the seen POI set-size has a direct impact on number of unseen objects in smart museums, which has effect on difficulty of predicting relevant POI in the one-shot POI recommendation problem. Similarly, we have studied

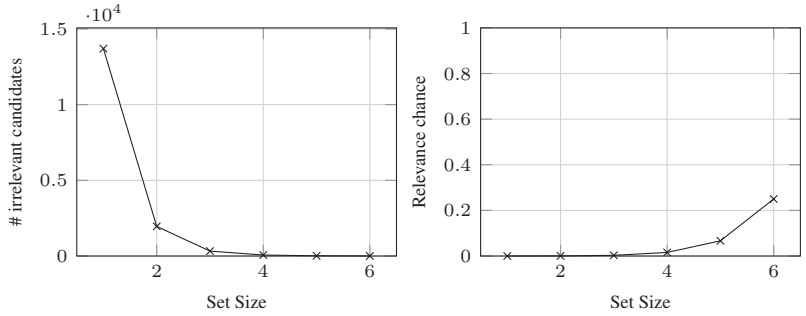


Figure 2.12: Impact of seen POIs set size on unseen POIs route recommendation based on number of irrelevant candidates in contrast to just one relevant route (left figure) and relevancy chance of a random recommended unseen POIs route (right figure).

impact of seen POI set-size on number of candidate routes, in which just one of the routes is the relevant one. Figure 2.12 shows number of candidate routes at each seen POIs set-size, which is calculated based on the following equation:

$$N_{rc} = \sum_{k=1}^{n} \frac{n!}{(n-k)!},$$

where k is the size of sequence of unseen predicted POIs, n is the number of unseen POIs in a user's session, and N_{cr} is the total number of route candidates having from one to n route length (number of POIs in the recommended route). As it is shown in Figure 2.12, due to the number of irrelevant routes available for each relevant unseen POIs route, the unseen POI route recommendation is a more challenging problem to address compared to the next POI recommendation problem. We leave investigation on this problem in smart environment to a future work.

One could easily think of further extensions of this, using the same kind of techniques to address related problems emphasizing different aspects. Of particular interest is to look the social aspects of smart exhibition visits, and ways to bring social aspects into the digital realm. A specific interesting problem to tackle here is recommending the most similar visitors, rather than items or object, in the smart environment. This could be a great strategy to bring the social aspect to museum visits. As it is discussed in [112], using mobile tour guides has negative social effects such as less interaction with visitors' fellow group members in a group visit. However, recommending similar users in a museum who are most likely take a same route and visit same objects, we can motivate individual users to create a group whose members have similar preference. In this way, we could have a positive impact on social aspect of museum visits, by showing the steps of prior, like-minded visitors, and bring the museum and the digital alive.

2.7 Discussion And Conclusions

The main focus of this section is the study of how to build a behavioral user model for the set-based POI recommendation problem using users' both onsite and online

information interaction behaviors. Our study on the strength of using each type of users' interaction behaviors with IoT in understanding users' onsite information interaction preferences shows that POI recommendation systems trained using features extracted from a combination of both onsite physical and online digital information interaction behaviors (i.e., online features) performs better than the ones trained by explicitly given context or onsite information interaction behavior. Therefore, we conclude that there is a similarity between onsite physical and online digital interaction preferences that causes an improvement on the onsite POI recommendation effectiveness.

Furthermore, we have studied the critical one-shot POI recommendation problem. According to our analysis, the learned models based on just basic explicit given contexts or onsite users' behaviors do not improve the hard-to-beat defined baselines (i.e., popularity and bias-based filtering). However, using a deep multilayer perceptron based on features extracted by online interaction behaviors leads to a significant improvement over the best baseline in all the defined evaluation metrics. Specifically, it has a statistically significant improvement over all baselines with 23% improvement in term of $p@1$ and 11% improvement in term of MRR. Therefore, our proposed approach is very effective in critical one-shot POI recommendation.

Furthermore, we have studied impact of seen objects set size on the performance of the proposed POI recommendation systems. According to our experiment, the recommendation performance is generally increased proportional to the seen object set size. Although external factors have impact on users' behavior at seen set size four in the exhibition, our proposed deep MLP model based on online features is less sensitive to the external factors and performs better than other models and baselines at all seen objects set sizes.

Our proposed MLP approach achieves 83% precision at rank 1 on the critical one-shot POI recommendation problem, realizing the high accuracy needed for fruitful deployment in practical situations. The proposed behavioral user model is generic and can be widely used in any environment with an integrated Internet of Things (IoT) infrastructure. Specifically, in the Cultural Heritage domain, the IoT applications hold the promise to provide a more interactive and multisensory experiences for visitors, and is expected to be integrated into museum practice in the next years [56, 96]. Our proposed model exploits online features hence is only applicable in cases where an online search engine with the similar objects or content related to the POIs is available for extracting the online features. Although many museums and organizations have a website with a search engine on their collection, it may not be the case in other applications in different types of smart environments.

Our general conclusion in this chapter is that it is possible to fruitfully combine information interactions in the online and physical world for effective recommendation in smart environments, thereby effectively blending real-world and online behavior in principled ways. This is an attractive direction, as IoT data is typically far more sparse than online data due to physical or geographical constraints on users requiring to be physically in the smart space.

This chapter addressed improving users experience in smart environments such as smart museums by modeling users interaction with smart devices. In the next chapter, we focus on creating reusable test collection for improving user experience in tourist

attraction recommendation domain in smart environments such as smart cities.

3

Test Collection Building for Contextual POI Recommender Systems

Providing effective personalized and contextualized POI recommender systems to users can lead to enhancing user experience in smart environments. To evaluate and improve personalized and contextualized POI recommender systems, a reusable controlled test collection can lead to flexibility of an effective offline evaluation of the personalized and contextualized POI recommender systems.

In this chapter, we address *RQ2: How to create a reusable test collection for the Contextual Suggestion problem?* We detail how we organize TREC contextual suggestion track with an aim of creating reusable test collection for the contextual POI recommendation problem.

3.1 Introduction

The TREC Contextual Suggestion Track ran for the fifth and last year as an independent track in 2016 [39–41, 43]. The track has the primary goal of providing reusable test collection for evaluation of point-of-interest (POI) recommendation systems. The test collection is open to anyone who is willing to do research in contextual suggestion problem.

The contextual suggestion track assumes a traveller in a specific context (e.g., a city and trip type) seeking things to do that reflects their own interests, which is supposed to be inferred from their interests in the given context and a visited city (seed cities in the track). Given a user's contexts and profile including a POI list, their tags/endorsements, and ratings from the seed cities, participants make recommendations for attractions in a new context (including the target city as the location).

For example, imagine a group of information retrieval researchers with a November evening to spend in beautiful Gaithersburg, Maryland. A contextual suggestion system might recommend a beer at the Dogfish Head Alehouse[1], dinner at the Flaming Pit[2], or even a trip into Washington on the metro to see the National Mall[3].

[1] www.dogfishalehouse.com
[2] www.flamingpitrestaurant.com
[3] www.nps.gov/nacc

The track has been operating since 2012, we discuss the final setup as run at TREC 2016. The main changes compared to earlier years were:

1. The track provides a fixed TREC Contextual Suggestion Web corpus as an additional data to overcome the dynamic nature of the open web.
2. The track provides endorsements (i.e., tags) of venues.
3. The track was split into two phases:
    3.1. Phase 1 experiment, which is a collection based task similar to the TREC 2015 Contextual Suggestion Track's Live Experiment. The main change is that the track does not require participants set up and register a live server. However, the track distributes a set of profiles and contexts and collect responses in a batch wise fashion, as was used in the track until 2014.
    3.2. Phase 2 experiment, which is a reranking task similar to the TREC 2015 Contextual Suggestion Track's Batch Experiment.
4. The track used a multilayer pooling approach that aimed creating a reusable test collection, which was very challenging in previous years of the track [72, 74] as is detailed in the next chapter.

The rest of this chapter is organized in the following way. Next, in Section 3.2, we will detail the track's tasks. This is followed by a discussion of the resulting test collection in Section 3.3 and the pooling method in Section 3.4. Then, Section 3.5 details the evaluation results of all submissions and teams. We conclude the chapter in Section 3.6.

3.2 Task Overview

This section will discuss the tasks of the TREC 2016 contextual suggestion track. The track followed the setup of 2015 with two distinct phases. In both phase 1 and phase 2 tasks, participants were asked to develop a system that is able to make suggestions for a specific person based on their given profile and context. As input of the task, the track organizers provide a set of profiles, a set of contexts and a set of example suggestions (URLs of pages corresponding to POIs in a given context). Each profile corresponded to a single user's preferences in example suggestions of another context or city, their gender and age, and each context includes information about the target city (i.e., the target location), a trip type, a trip duration, a type of group the person is travelling with, and a season the trip will occur in.

Profiles correspond to the stated preferences of real individuals, who either recruited through crowdsourcing or recruited editorial judges. These assessors first judged example attractions in seed locations, later returning to judge suggestions proposed by the phase 1 participants for various contexts. Both for the profile (i.e., seed pages) and for the suggested recommendations, assessors were able to choose the context or city for which recommendations were judged.

As output of the phase 1 task, for each context/profile pair, participants were required to return a ranked list of 50 suggestions. Each suggestion was expected to be relevant

Table 3.1: TREC Contextual Suggestion track collection example.

Attraction ID	City ID	URL
TRECCS-00000005-418	418	http://www.greatfallsmt.net/people_offices/park_rec/gibson.php
TRECCS-00000006-418	418	http://www.mackenzieriverpizza.com
TRECCS-00000007-418	418	http://www.bostons.com
TRECCS-00000008-418	418	http://pink.victoriassecret.com

to the given profile and the context. As output of the phase 2 task, participants were expected to rerank the given suggestion candidates with respect to the user's profile and context and return them as the phase 2 response. To be precise:

Phase 1 Experiments The phase 1 experiment is a collection based task, in which participants are asked to develop a contextual suggestion system that is able to make suggestion for a particular person in a specific context. In particular, for each given request (including profile and context), participants has to retrieve 50 suggestions from the TREC contextual suggestion collection as a response.

Phase 2 Experiments The phase 2 experiment is a reranking task, in which a suggestion candidates set is provided for each request. In fact, all the suggestion candidates available in phase 2 requests were made by participants in phase 1. Therefore, we have all the judgments of the suggestions available in the suggestion candidates, which facilitates the reuse of the contextual suggestion test collection.

The track continues to use a collection of URLs corresponding to POIs in each context that was released in 2015, see the examples in Table 3.1. For the future studies on the contextual suggestion problem using the TREC contextual suggestion track qrels, due to the dynamic nature of the collection, we strongly recommend to use the TREC Contextual Suggestion Web corpus, which will be introduced in Section 3.3.2.

3.3 Test Collection

This section discusses the resulting test collection. TREC 2016 contextual suggestion test collection consists of a corpus (including TREC contextual suggestion collection and the web corpus), a set of requests, and relevance judgments. In addition we have also released suggestions' endorsements.

3.3.1 TREC CS Collection

The TREC Contetxual Suggestion collection was collected by asking participants as volunteers to retrieve suggestion candidates related to each city from the open web in a pre-task phase. This collection was created in TREC 2015 contextual suggestion track. The collection consists of a set of attractions. For each attraction there are:

1. An attraction ID, which contains three parts separated by dashes (-)
    1.1. The string 'TRECCS'

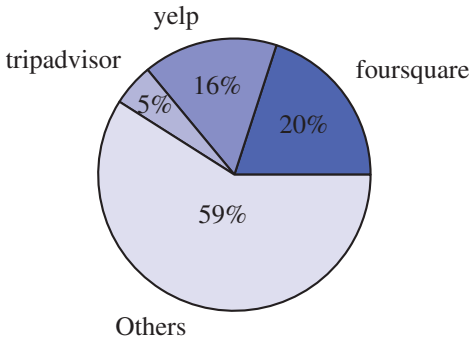


Figure 3.1: Most popular domains in the TREC Contextual Suggestion Web Corpus.

   1.2. An 8 digit number

   1.3. A three digit number corresponding to that attraction's city ID

2. A city ID which indicates which city this attraction is in

3. A URL with more information about the attraction

4. A title

An example of the TREC Contextual Suggestion collection is given in Table 3.1.

3.3.2 TREC CS Web Corpus

In addition to the TREC contextual suggestion collection, which is available since 2015, we released TREC contextual suggestion web corpus. The TREC CS web corpus is a web crawl of the suggestions' URLs available at the TREC contextual suggestion collection. In this crawl, we have managed to fetch 77.39 % of the whole TREC Contextual Suggestion collection, which is 956,437 web pages out of 1,235,844 URLs.

This crawl includes web pages from different domains like yelp, tripadvisor and foursquare. Yelp was the most difficult domain to crawl, and we managed to crawl about 153K out of 220K yelp web pages available in the TREC contextual suggestion collection. Figure 3.1 indicates percentage of available POIs from the most popular tourist attraction domains in the TREC Contextual Suggestion Web corpus. As it is shown in this figure, Foursquare, Yelp and Tripadvisor are the most popular domains in the TREC Contextual Suggestion Web corpus.

The TREC Contextual Suggestion Web Corpus includes attraction web pages of 272 different North American cities. In this corpus, there are 3,516.31 tourist attraction web pages in average per city. The corpus is in a WARC (Web ARChive) format. In order to have access to the data designated as the TREC CS Web Corpus, organizations must first fill in a data release Organizational Application Form. Then, the signed form must be scanned and sent by email to data@list.uva.nl. On receipt of the form, participants will be sent information on how to download the corpus.

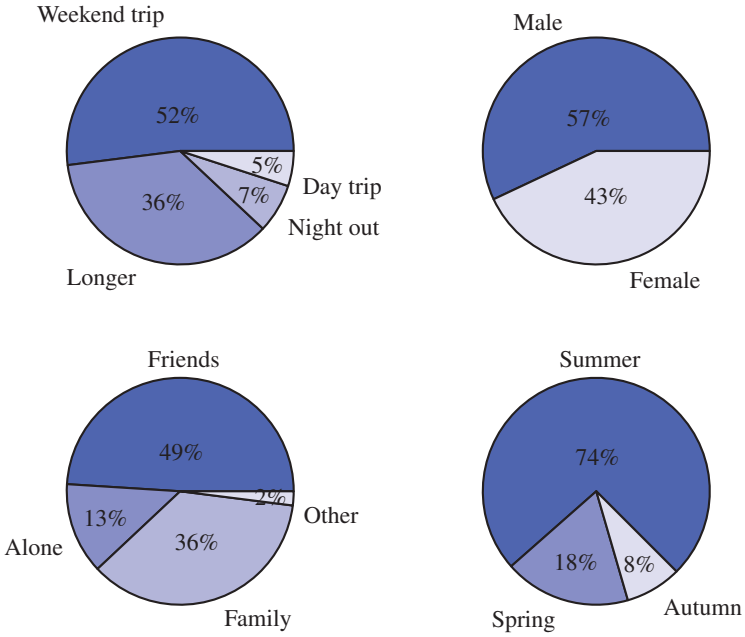


Figure 3.2: Example of official phase 1 requests' contexts and profiles statistics.

3.3.3 Requests

In both phase 1 and phase 2 experiments, each request contains information about assessors' preferences as profiles and their chosen context. Moreover, phase 2 requests contains suggestion candidates related to each profile and context pair. Each profile consists of a list of attractions the assessor has previously rated, their gender and their age. For each attraction the profile will include:

1. A rating:
    1.1. 4: Strongly interested
    1.2. 3: Interested
    1.3. 2: Neither interested or uninterested
    1.4. 1: Uninterested
    1.5. 0: Strongly uninterested
    1.6. -1: Not loaded or no rating given
2. Tags/endorsements if it is applicable.

Each context consists of a city name which represents which city the trip will occur in and several pieces of data about the trip. The context is as follows:

1. A city the trip will occur in (e.g., Seattle)
2. A trip type (e.g., Business)

```
{"id":743,
"body": {
    "group": "Friends",
    "season":"Summer",
    "trip_type":"Holiday",
    "duration":"Weekend trip",
    "location":{
        "state":"TX",
        "id":306,
        "name":"Waco",
        "lat":31.54933,
        "lng":-97.14667},
    "person": {
            "gender": "Male",
            "age": 28,
            "id": 15012,
            "preferences":[
                {
                "rating":4,
                "documentId":"TRECCS-00211395-161",
                "tags":[
                     "Cocktails",
                     "Restaurants"
                     ]},
                ...
                ]
                }},
"candidates":[
    {"documentId":"TRECCS-00267253-306",
    "tags":[
        "Family Friendly",
        "Restaurants"
        ]},
    {"documentId":"TRECCS-00294259-306",
    "tags":[
    "Entertainment",
    "Live Music"
    ]},
     ...
    ]
    }
```

Figure 3.3: TREC Contextual Suggestion Track phase 2 request example in JSON format

3. A trip duration (e.g., Weekend trip)

4. A type of group the person is travelling with (e.g., Travelling with a group of friends as "Friends")

5. A season the trip will occur in (e.g., Summer)

An example of the TREC Contextual Suggestion phase 2 request is shown in Figure 3.3. The track organizers provide 438 input requests in total, in which requests having identifiers from 700 to 922 are used for the official experiments in TREC 2016 contextual suggestion track. In particular, TREC 2016 Phase 1 test collection consists of judgments of 61 requests, and TREC 2016 Phase 2 test collection includes all the phase 1 requests except requests having 707, 912 and 922 as identifiers, hence 58 requests in total. The difference is a result of some additional judged requests coming available after the release of the phase 2 requests. Some examples of official phase 1 requests' context and profile statistics are shown in Figure 3.2.

In building profiles for the TREC 2016 official requests (request IDs $>= 700$), two seed cities were chosen (Seattle and Detroit). Each seed city had 30 POIs to be judged as user profiles. Users could choose which seed city to judge. If they just rate POIs of one of the cities, their profiles have 30 rated POIs. If they rate both of the seed cities' POIs, their profiles have 60 rated POIs. For example, in Phase 2 official requests, there are 39 requests having 30 judged example suggestions and 19 requests having 60 judged example suggestions in their profiles.

In phase 2 requests, due to the use of multi-depth pooling, which will be detailed in Section 3.4, the size of provided suggestion candidates is varied per request. Specifically, average number of suggestion candidates over the 58 phase 2 requests is 96.53, maximum number of suggestion candidates is 119 and minimum number of suggestion candidates is 79.

The rest of the requests, which were collected in TREC 2015, were used as train set of the TREC 2016 contextual suggestion track, as the qrels of those requests were available since TREC 2015. The TREC 2016 identifiers of those requests are same as the one used in TREC 2015, which facilitates evaluation of these requests based on the TREC 2015 contextual suggestion test collection. However, we have created a new pool and new sets of suggestions as suggestion candidates using the multi-depth pooling approach, which will be discussed in Section 3.4. Therefore, suggestion candidates of those requests available in TREC 2015 are different from the ones in TREC 2016. In fact, TREC 2015 batch requests contain a set of suggestion candidates with a very high probability of being relevant to the request. To make it a more realistic and challenging problem, we have injected more noise into the original batch requests of TREC 2015, hence the sets of candidates for the 2015 requests included this year differs from those of last year.

There are further requests that are based on requests made during the TREC 2015 live tasks. There were left out of the TREC 2015 data, privileging only a single request per crowdsourced assessor, but judgement are available to be used. As these requests were not as deeply pooled as the official TREC 2016 requests, they are excluded again from the official test collection in 2016, but may be released separately at a later date.

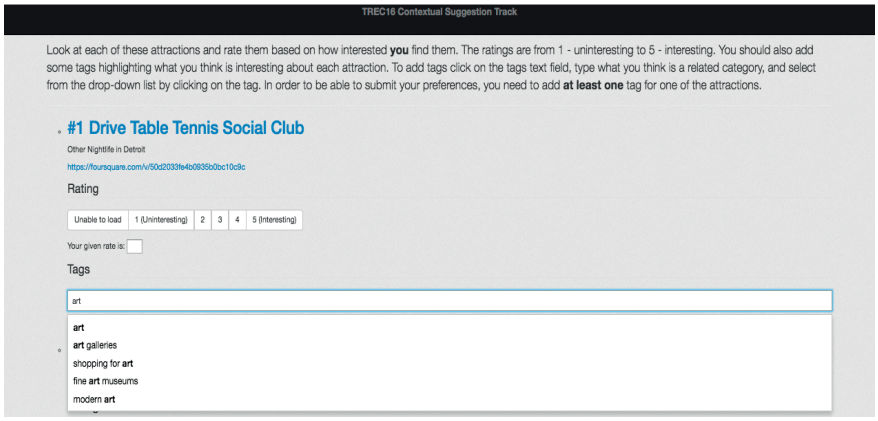


Figure 3.4: An example of how assessors give rating and tags/endorsements to the suggestions.

3.3.4 Relevance Judgments

Relevance judgments were collected through crowdsourcing and by the help of a group of graduate students. They were asked to rate suggestions in a same scale that presented in Section 3.3.3.

However, in the qrels, we have shifted the raw assessors' 5 point scale judgments with -2, making the judgments in the range -3 to 2, and making a score of 1.0 or higher correspond to a "interested" or "strongly interested" judgment. Therefore, the trec_eval can be used to evaluate contextual suggestion runs based on all the common IR measures, included graded measures like NDCG.

3.3.5 Suggestions Endorsements

In addition to the relevance judgments based on the ratings, we also asked the assessors to endorse the suggestions using the tag field, which is shown in Figure 3.4.

In practice, endorsement was not an easy task for them, and they were not willing to give tags to all the given suggestions. Therefore, NIST assessors endorsed all the pooled suggestions, and we include those tags/endorsements to both profiles and suggestion candidates of the phase 2 requests.

3.4 Pooling Approach

This section discusses the pooling approach used at the TREC 2016 contextual suggestion track. Previously, TREC contextual suggestion organizers used the traditional pooling approach and pooled all the top-N suggestions of the submissions, in which N is a pool cut-off. They created a pool using 5 as the pool cut-off. According to the studies done on the reusability of the TREC contextual suggestion test collection [67, 72, 74, 75], reusability of the test collection suffered a lot from the personalization effects and respectively the shallow pool cut-off. To address this issue, we experimented with a "multi-depth" pooling approach.

3.4.1 Multi-Depth Pooling

In the multi-depth pooling approach, in addition to the pool cut-off (hard pool cut-off), we define two others pool cut-offs, namely, soft pool cut-off and very soft pool cut-off. In the multi-depth pooling approach, we have pooled the following suggestions:

1. All the suggestions/documents ranked higher than the hard pool cut-off by any of the submissions is pooled. This would guarantee an stable measures up to the traditional pool cut-off.

2. In addition, if a suggestion/document ranked higher than the soft pool cut-off by at least one submission, and also ranked higher than the very soft pool cut-off by at least one run from another participated team, the suggestion is pooled. This would have effects on having more stable measures deeper than the traditional hard pool cut-off in the ranking.

Following last years of the TREC contextual suggestion track, we have used 5 as the hard pool cut-off. In addition, taking into account the effort needed to create the test collection, we have set 25 as the soft pool cut-off and 50 as the very soft pool cut-off as this leads to a pool size of about 100 suggestions per request.

The proposed pooling approach would give us more stable evaluation results over deeper ranks than the traditional pool cut-off. The traditional pooling approach with 5 as the pool cut-off would cost 3,377 judgments for the 61 official phase 1 requests. Interestingly, the above multi-depth pooling approach spend even less effort than pooling top-10 documents/suggestions provided by the submissions. Specifically, for the official qrels of the TREC 2016 contextual suggestion, we have collected 5,898 judgments using multi-depth pooling approach, in which we have got 5,782 official judgments after filtering some noises. If we had used the traditional pooling approach with 10 as the pool cut-off, we would have collected 6,206 judgments.

3.4.2 Fraction of Judged Documents

In multi-depth pooling, we have pooled deeper and expected a larger fraction of judged documents after the pool cut-off. Figure 3.5 shows a comparison of the cumulative overlap@N [72] in TREC 2015 and 2016 Contextual Suggestion tracks. As it is shown in Figure 3.5, the fraction of judged documents gently decreases after the hard pool cut-off (i.e., 5) in TREC 2016 contextual suggestion test collection. However, in TREC 2015 contextual suggestion track, fraction of judged documents dropped dramatically after the pool cut-off (i.e., 5). We have also plotted just-in-rank overlap@N in Figure 3.5, in which we just consider fraction of judged and unjudged documents at rank N and calculate the overlap. This figure indicates that the multi-depth pooling is effective in minimizing the fraction of unjudged documents in ranks deeper than the pool cut-off. The larger fraction of judged documents in TREC 2016 helps us to have a more stable evaluation over ranks deeper than the traditional pool cut-off.

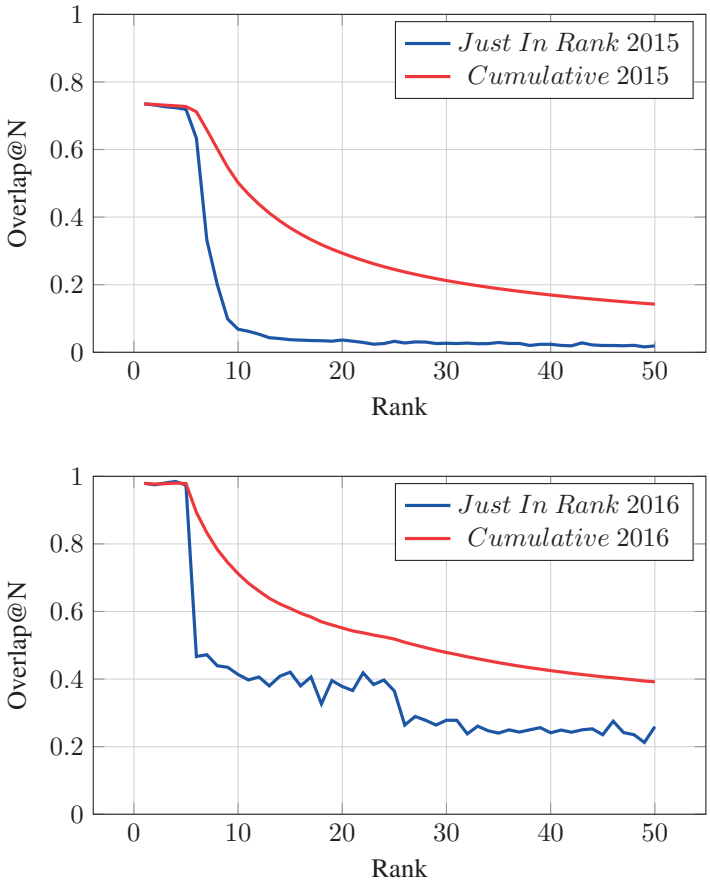


Figure 3.5: Cumulative and just-in-rank Overlap@N in TREC 2015 and 2016 contextual suggestion test collections.

3.4.3 Reusability

In this section, we study reusability of the phase 1 TREC contextual suggestion test collection [4]. As shown in Figure 3.5, the fraction of judged documents has improved in ranks deeper than the hard pool cut-off using multi-depth pooling. However, effects of this improvement on the reusability of the test collection are not a priori clear.

Figure 3.6 demonstrates reusability of the TREC 2016 contextual suggestion ranking (phase 1) test collection based on Leave-One-Team-Out (i.e., LOTO) [22] test. According to Figure 3.6, the TREC 2016 contextual suggestion phase 1 test collection should be used with some care based on P@5 metric. The official runs are completely judged up to rank 5, by design of the pooling approach, but post-submission exper-

[4]The next chapter addresses a more detailed analysis of the TREC contextual suggestion track test collection and its reusability.

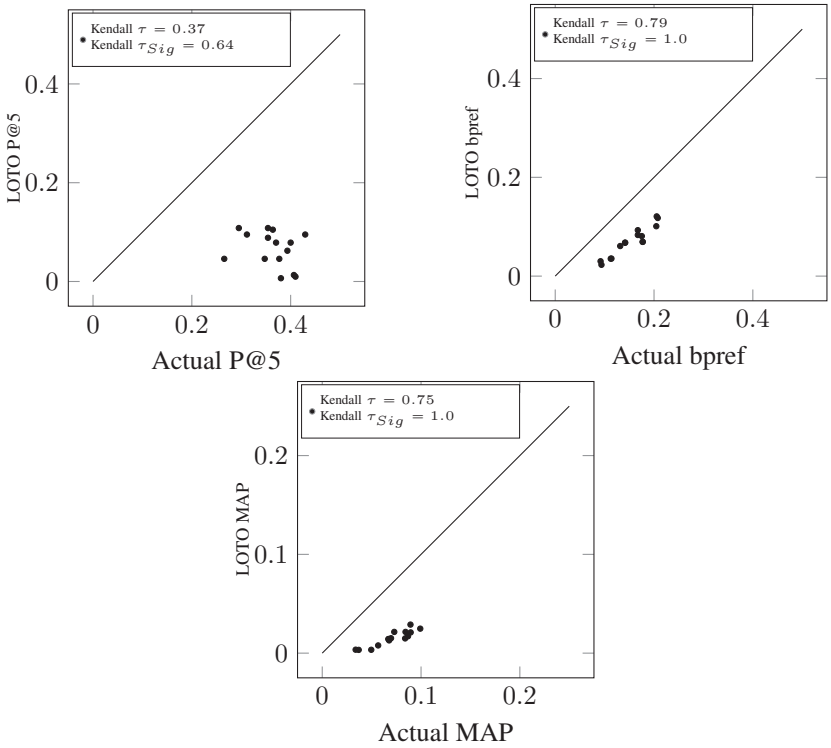


Figure 3.6: Leave One Team Out (LOTO) reusability test of the contextual suggestion test collection created based on multi-depth pooling.

iments not contributing to the pool of judged documents risk being underrated. We have observed a similar system ranking correlation based on NDCG@5 metric having Kendall's $\tau = 0.43$.

There is also good news: the phase 1 test collection appears to be reusable when considering the more stable evaluation measures for incomplete test collections. Specifically, the test collection has got perfect system ranking correlation between official TREC system ranking and the LOTO system ranking based on the Kendall's τ using statistical significant inversions using MAP and bpref metrics. In this test, 54% of the pairwise comparisons are significant based on MAP and we have had 64% significant differences based on bpref.

Furthermore, the TREC 2016 contextual suggestion phase 2 test collection is perfectly reusable by design as it is created for the contextual suggestion reranking and all relevance judgments of suggestion candidates are available in the test collection.

3.5 Evaluation Results

In this section, we first list our official evaluation measures. Then, we detail the evaluation results of phase 1 and 2 experiments.

Table 3.2: Official TREC 2016 Contextual Suggestion Track's *phase 1 submissions* of top-5 teams in the ranking evaluated over 61 requests.

Rank	RunID	**NDCG@5**	P@5	MRR	MAP	bpref
1	USI2	*0.2826*	*0.4295*	0.6150	0.0868	0.1772
2	IAPLab1	0.2789	0.3770	*0.6245*	0.0729	0.1672
3	ADAPT-TCD-r1	0.2643	0.4066	0.5777	*0.0992*	0.2046
4	FUM-IRLAB-3	0.2601	0.3803	0.5824	0.0566	0.1124
5	FUM-IRLAB-1	0.2596	0.4000	0.5501	0.0696	0.1672
6	ADAPT-TCD-r2	0.2595	0.4098	0.5512	0.0895	0.1753
7	USI1	0.2578	0.3934	0.6139	0.0839	0.1769
8	FUM-IRLAB-2	0.2544	0.3705	0.5945	0.0677	0.1315
9	ExPoSe-response-tags	0.2461	0.3639	0.5206	0.0496	0.1138
10	ExPoSe-response-all	0.2445	0.3541	0.5128	0.0672	0.1413
11	ExPoSe-response-content	0.2443	0.3541	0.5114	0.0669	0.1416

3.5.1 Evaluation Measures

Three measures are used to rank both phase 1 and phase 2 runs. Our main measure is NDCG@5; in addition, P@5 and MRR are also used as two other metrics have been used since 2012 in TREC contextual suggestion track. As early rank cut-off measures are notably unstable, we also include measures taking more of the ranking into account, such as P@10, NDCG, MAP, Rprec and bpref, also profiting from the deeper pooling approach of this year.

The official results for the phase 1 task are shown in Table 3.2. The best phase 1 runs from top-5 teams out of 8 participated teams in phase 1 will be detailed in Section 3.5.2. Table 3.3 shows the official results for the phase 2 task. The best phase 2 runs from top-5 teams out of 13 participated teams in phase 2 will be summarized in Section 3.5.3.

3.5.2 Best Performing Phase 1 Submissions

The five best performing teams in the phase 1 evaluation are the following:

USI

USI [4]'s best performing phase 1 run is "USI2", in which they crawled Foursquare for virtually 600K venues. Using the crawled data, they created positive and negative category profiles consisting of all categories a user liked/disliked as well as their corresponding normalized frequencies. The initial category profiles are then used to measure the similarity between a new venue and a particular user. They created the initial ranking and picked the top 10 venues for each user to gather extra information about them. For each user they also created positive and negative frequency-based venue taste keyword profiles. For the new set of venues, they extracted venue taste

keywords and measured the similarity between the venues and a particular user. They reranked the top 10 venues for each user in the initial ranking using a linear combination of the venue category and taste keyword scores.

IAPLab

Nanjing University's IAP Lab did not provide a description of their approach by the time of writing, nor submitted a participants' paper to the TREC Notebook or TREC Proceedings. Therefore, we cannot provide a further description of their approach in the overview paper, apart from noting that their system did well for the phase 1 task.

ADAPT_TCD

ADAPT_TCD [16] proposed an ontology-based approach, using an ontology that was constructed using the Foursquare Category Hierarchy. The three models, each based upon this ontology, are: User Model, Document Model and Rule Model. For the User Model they build two models, one for each phase of the task, based upon the attractions that were rated in the user's profile. In the first phase they use only the positively rated attractions from each user. In the second phase they use both positive and negatively rated attractions to build the user model. The Document Model enriches documents with extra metadata (tags) from Foursquare and categories (concepts) from the ontology are attached to each document. The Rule model is used to tune the score for each candidate suggestion based upon the context of the trip and how it aligns with the rules in the model.

Their best performing run is "ADAPT_TCD_r1" in which, they build the user positive model based on the positively rated attractions in the user's profile. For each of these attractions, they create an index of all the classes, based on Foursquare data, that these attractions are an instance of, along with the tag set that was found on that attraction's page on Foursquare. They then compute the count per class and then the percentage of each class in the positive model. For a given place p that a user is travelling to, they select the documents that match the classes in the positive model. They eliminate the documents that belong to a class that violates at least one rule in the rule model. They retain the class percentage breakdown from the user model and map these percentages to 50 and represented this as a number, x, for each class. Following this, they select the top x attractions of this class from the retrieved documents after ranking them based on the features that have been collected in the Document Model from Foursquare, which are: the average users' rating, the users' rating count, the users' reviews count and the tag similarity measure between a document's tag set and the class tag set. After they select the required number of documents for all classes in the user model, they start to rank the documents based on the first three features mentioned before and return the final ranked list. If the number of attractions belonging to a specific class, in a specific city, do not meet the required number, they compensate for the shortfall by getting more attractions from the highest ranked class/classes in the user model.

FUM-IRLAB

FUM-IRLAB [104] followed two main approaches for finding suitable attractions for a given user: a content-based approach and a category-based approach.

In the content-based approach, all Web pages related to attractions are modeled as vectors of real numbers using word embedding and document embedding techniques. Then, similarities between attractions in the profile of a given user and new attractions are calculated using methods for finding similarities between vectors.

In the category-based method, a subset of attractions is modeled as a vector of categories. These categories are extracted from the category information of the related Yelp, TripAdvisor, or Foursquare pages of the attractions. In addition, a user profile is modeled as a vector of categories, where these are categories extracted based on a mapping from the tags provided in the user's profile and the categories extracted for the attractions. Finally, similarities between attractions and user profiles are calculated based on similarities between these vectors. They submitted three methods of combining these two approaches to this track as three different runs.

Their best performing run is "FUM-IRLAB_3", in which the document-embedding vectors and the similarities between them are employed to produce a list of the most similar attractions to each attraction in the user profile. They found that despite a lot of very related results, this list contains a couple of completely unrelated pages. Hence, they decided to filter the result set for having a more precise list of attractions. They made an intersection between these lists with the attractions provided by category-based approach, making them more precise in the cost of decreasing recall. For each liked attraction in the user profile, they created a list of similar attractions, and then they iteratively selected two top attractions from each list and merged them to the final result set. They continue their iterations until they find 50 results from these lists.

ExPoSe

ExPoSe [44] focused on one of the key steps of contextual suggestion methods is estimating a proper model for representing different objects in the data like users and attractions. They used the Significant Words Language Models (SWLM) as an effective method for estimating models representing significant features of sets of attractions as user profiles and sets of users as group profile. The SWLM model outperformed the standard language model, and is robust against negative examples.

For phase 1, the tag based run "ExPoSe_response_tags" obtained a better score than the content-based, and the combined run—although the differences between the runs were small.

3.5.3 Best Performing Phase 2 Submissions

The five best performing teams in the phase 2 evaluation are the following:

DUTH

DUTH [98] have further developed and built upon the two methods they first presented in Contextual Suggestion 2013, which they have fine-tuned using TREC 2015 data.

Table 3.3: Official TREC 2016 Contextual Suggestion Track's *phase 2 submissions* of top-5 teams in the ranking evaluated over 58 requests (excluding 707, 912, 922).

Rank	RunID	**NDCG@5**	P@5	MRR	MAP	bpref
1	DUTH-rocchio	*0.3306*	0.4724	0.6801	0.4497	0.4704
2	Laval-batch-3	0.3281	*0.5069*	0.6501	0.4536	0.4666
3	USI5	0.3265	*0.5069*	0.6796	0.4590	0.4507
4	DUTH-bcf	0.3259	0.4724	0.5971	*0.4606*	*0.4845*
5	USI4	0.3234	0.4828	*0.6854*	0.4576	0.4494
6	Laval-batch-2	0.3118	0.4345	0.6287	0.4378	0.4721
7	DUTH-knn	0.3116	0.4345	0.6131	0.4456	0.4825
8	bupt-pris-2016-cs.2-.4-max	0.2936	0.4483	0.6255	0.4318	0.4476
9	Laval-batch-1	0.2889	0.4276	0.6372	0.4397	0.4409
10	UAmsterdamDL	0.2824	0.4448	0.5924	0.4168	0.4452
11	bupt-pris-2016-cs.4-.2-max	0.2761	0.4241	0.5937	0.4308	0.4465
12	UAmsterdamCB	0.2730	0.4069	0.5631	0.4076	0.4337

They address the task by individually using two classification methods, namely, a weighted k-NN classifier and a modified Rocchio classifier. Also, as a third method, they explore the use of election systems, namely Borda Count, as a means of fusing the results of the two aforementioned classifiers.

Their best performing run is "DUTH_rocchio", which is based on a Rocchio-like classifier. Using a user's rated venues as training examples, they build a custom query for the user using a modified Rocchio relevance feedback method. Specifically, they build a centroid per rating and combine/add those using their corresponding ratings as contributing factors, offset by 2 so as ratings 0 and 1 provide negative feedback with -2 and -1 weights respectively. Rating 2 is eliminated as neutral.

LavalLakehead

LavalLakehead [124] formulate a customized query according to user profile to retrieve the 100 initial attractions. Then these 100 candidates are ranked by two independent ranking models who cover global trend of interests and contextual individual preference respectively. The first model is a pre-trained regressor on 2015 TREC data thus it can prioritize popular places and categories loved by all users (E.g. Museums and National Parks). The second model introduces word embedding to captures individual user preference. Both user profiles and candidate places are represented as word vectors in a same Euclidean space. So that a similarity score between user and attraction can be calculated by measuring their vector distance. In the end, a final ranking is given by summing up the two models' scores, and "Laval_batch_3" is a result of the combination of the two above models.

USI

USI [4]'s best performing phase 2 run is "USI5", in which they computed a set of multimodal scores from multiple locationbased social networks (LBSNs) and combined them with a score that predicts the level of appropriateness of a venue to a given user context. Briefly, the scores are calculated as follows: positive and negative reviews are used to create user profiles to train a classifier which then predicts how much a particular user will like a new venue. Moreover, the frequency-based scores are calculated based on the venue categories and taste keywords. As for the prediction of appropriateness, they created two datasets using crowdsourcing and trained a classifier with the features they extracted from the datasets. A linear combination of all the scores produced the final ranking of the candidate suggestions.

bupt_pris_2016

BUPT [174] collected data by crawling from the Yelp API and Foursquare API. With attractions marked with rating and tags in the preference list, they calculated users' average rating for each tag. For tags without a rating of the user in the profile, that is, the missing ratings, they filled them by Collaborative Filtering. Next, they got the users' rating for an attraction with either a mean function or a max function. By ranking the ratings of candidates, they git a ranked list for each user.

Their best performing run is " bupt_pris_2016_cs.2_.4_max", in which they put a higher weight on ratings from Foursquare (0.4), a lower weight on ratings from Yelp (0.2), and used a max function to calculate the users' rating for attractions.

UAmsterdam

UAmsterdam [78] studied contextual suggestion problem through neural user profiling and neural category preference modeling by the help of suggestions' endorsements being released by the TREC 2016 contextual suggestion track organizers. Their best performing run is "UAmsterdamDL", in which they studied how to predict relevant suggestions to the given user and context using category preference models.

In UAmsterdamDL, they cast the context-aware recommendation problem to a binary classification problem. In order to learn a user preference model, they have used a deep neural network with 4 hidden layers having 478 units, in which 123 suggestion-category relevance features have been used as inputs of the network. In this model, for each user, preferences in the user's profile considered as a train set and suggestion candidates available in the phase 2 requests considered as the test set.

3.6 Conclusions

This section concludes our overview of the TREC 2016 contextual suggestion track. The track's main aim is the creation of a reusable test collections for the personalized POI recommendation task, which has proved a difficult task according to the previous studies [72, 74]. To this aim, we released the TREC CS web corpus, which is a crawl of the TREC contextual suggestion test collection. By fixing the test collection's content, we

can overcome the dynamic nature of the contextual suggestion collection, and separate this effect from the personalization effects. We have also used a multi-depth pooling approach to improves reliability of the contextual suggestion systems scores based on measures at ranks deeper than the traditional pool cut-off. Moreover, we released attractions' endorsements being collected by NIST assessors, and participants showed considerable interest in using the endorsements to improve their contextual suggestion systems.

This chapter addressed creating a reusable test collection for evaluation of contextual suggestion systems in a smart city context. In the next chapter, we focus on maintaining and improving reusability of dynamic test collections such as the TREC contextual suggestion track test collection created in this chapter.

4

An Analysis of Test Collection Building in Dynamic Domains

As reusability of test collections in dynamic domains may degrade over time, test collections in dynamic domains may need maintenance with an aim of improving reusability of the test collection. In this chapter, we address *RQ3: Can we build a reusable test collection for a dynamic domain by injecting judged documents into a test collection with sparse judgments?* We study reusability of the TREC contextual suggestion test collection. We also show how to expand test collections with an aim of improving their reusability and how to test reusability of expanded test collections.

4.1 Introduction

Evaluation in Modern Information Retrieval (IR) tasks based on creating test collections is under threat by different factors such as presenting new tasks and new types of data. All recent IR research agendas [5, 23, 37, 99] seek ways to embrace these new challenges, while still retaining the advantages of experimental control in the Cranfield/TREC paradigm [33]. One particular challenge is to deal with the dynamic nature of the web and other online sources [147]. Apart from the challenges for assessors in judging dynamic collections in ways that reflect how real searchers would experience them while doing the task [162], the dynamic nature of collections contributes to test collections becoming dated or less representative for the evolving online behavior. Within years, months, or days, many web pages pooled during test collection building phase change, become irrelevant, or even disappear. There are also plenty of new web pages emerging that were not available during test collection building time, but they have been created after the test collection building. This is a common problem in web archiving, in which researchers put lots of effort in reconstructing and retrieving unarchived web pages from the web [91]. Many of these pages are relevant to the test collection tasks and should be added to the test collection as well as data collection in order to keep the test collection up-to-date.

The problem of maintaining test collections in dynamic domains with an aim of creating reusable test collections and maintaining their representativeness is of central importance as many IR evaluation forums' (e.g., TREC, CLEF, NTCIR, INEX and FIRE) test collections can become dated or even outdated due to the dynamic nature

of the collections. For example, many academic papers on web search still rely on test collections based on the VLC (1997), .GOV (2002), .GOV2 (2004), ClueWeb (2009), ClueWeb2 (2012) collections, which are useful in their own right but may fail to represent crucial aspects of rapidly evolving modern online search. These limitations of test collections are broadly known, but rarely discussed, and this chapter attempts to explicitly study some of these limitations in a particular case, explore ways to quantify their effects on comparative system evaluation, and experiment with some simple approaches that may help mitigate some of the limitations. This is by no means a magic wand or silver bullet that will resolve these hard and fundamental challenges. Rather, our general aim is to promote critical, reflective analysis of the test collections we build under these hard conditions, hoping to inform researchers using these test collections about the conditions under which they can be used with reasonable trust in their reliability, but also flag an appropriate call to caution when not.

This chapter is motivated by the TREC Contextual Suggestion track, investigating search techniques for complex information needs that are highly dependent on context and user interests [156]. It offers a personalized venue recommendation task based on a U.S. city as context, and crowdsourced profiles and judgments. The track suffered from the delayed availability of the ClueWeb12 collection, and decided to use no static corpus of documents but accept any page on the web in 2012. In the following years in TREC 2013 and 2014, the track used ClueWeb12 (consisting of 733,019,372 English web pages) but kept on allowing open web results by popular request of the track's participants. This unique setup of the contextual suggestion track leads to two distinct sets of judgments: one set consists of judgments of documents contributed by open web runs, and the other one includes judgments of ClueWeb12 documents provided by ClueWeb12 runs [40].

This fact raises several questions: Is the open web-based test collection, which includes the majority of the judgments (i.e., 25 out of 31 pooled runs in 2014), reusable? Is the ClueWeb12-based contextual suggestion test collection using a fixed corpus reusable? If not, is it possible to reuse the open web judgments to build a new corpus in order to create a more reusable test collection? We study the following research questions:

1. *How reusable is the OpenWeb and ClueWeb12 test collections of the TREC contextual suggestion?*
    - 1.1. *How reusable is the test collection for evaluating non-pooled systems?*
    - 1.2. *What is the fraction of judged documents?*
    - 1.3. *What is the impact of personalization on the fraction of judged documents?*
2. *How to expand a test collection in order to improve its reusability?*
3. *How reusable is the expanded test collection containing judged open web documents?*
    - 3.1. *How reusable is the expanded test collection for ranking systems?*
    - 3.2. *Are retrieval models able to retrieve the judged open web documents?*

This builds on our earlier work in [72, 74], where we found that both the Open Web and ClueWeb12 test collections of the TREC Contextual Suggestion Track have low reusability, due to a very low fraction of judged documents beyond the pooling cut-off, which is in turn due to the personalized setup leading to a low pool depth due for each context and profile pair. Yet, merging relevant document from the open web runs into the test collection—a so-called Easter Egg Hunting approach—can improve the fraction of judged documents up to the point that the expanded test collection is reusable based on standard LOU reusablity tests.

Although reusability of the TREC Contextual Suggestion test collection is improved by Easter Egg Hunting approach based on LOU test, the result of the test is questionable as none of the runs used in the reusability test of the expanded test collection is pooled. In fact, we need a new reusablity test to evaluate reusability of maintained dynamic test collections or test collections not created based on pooling. To this aim, we have simulated pools with different pool depth and done experiments to see what would have happened if a run had contributed to the pool of judged documents. So, in addition, we study the following research questions:

4. *What is the impact of simulated pooling on the reusability of the expanded test collection?*

    4.1. *Does non-pooled system ranking change by adding more simulated judgments?*

    4.2. *What is the most effective pool cut-off based on the simulated test collection?*

    4.3. *How to estimate simulated pooling bias of the test collection in ranking non-pooled runs?*

5. *How reusable is the expanded test collection for ranking pooled systems?*

    5.1. *How reusable is the personalized expanded test collection for ranking simulated pooled systems?*

    5.2. *What is the impact of personalization and pool depth on the reusability of the test collection?*

The rest of this chapter is organized as follows. In Section 4.2, we review some related work on reusable test collection building and reusability tests. Section 4.3 is devoted to reusability evaluation of the Open Web and ClueWeb12 test collections. Our proposed test collection building approach is detailed in Section 4.4. The created test collection reusability is thoroughly evaluated in Section 4.5 for non-pooled runs. Then, we detailed simulated pooling and our proposed leave-uniques-in test in Section 4.6. Section 4.7 includes experimental result of the expanded test collection reusability based on the novel leave-uniques-in test for simulated pooled runs. Finally, we present the conclusions, discussion and future work in Section 4.8.

4.2 Related Work

In this section, we will discuss related work on test collection construction, pooling, reusability and related research in the TREC contextual suggestion track.

4.2.1 Test Collection Building and Pooling

At TREC, it is common to use the classical Sparck Jones and Van Rijsbergen [151] pooling technique by the National Institute of Standard and Technology (NIST) in order to create test collections for the comparative evaluation of retrieval systems. The idea behind pooling is that documents retrieved by a run in ranks deeper than the pool cut-off, is likely retrieved by another run inside the pool. The reusability of the resulting test collection depends on the completeness of the relevance judgments. Therefore, identifying an effective pool depth for building reusable test collections become an important issue. To this aim, Zobel [186] studied effects of pool depth on the reusability of test collections that low pool depth tends to lessen reusability of the test collections.

Within the literature on building a reusable test collection based on the pooling technique, one approach is to sample a more effective set of documents as a pool of documents to be judged. Cormack et al. [36] proposed iterative searching and judging technique, in which to retrieve and judge the highest possible number of relevant documents for each topic, assessors perform multiple searches in documents' relevance assessments process.

Moreover, due to the cost of test collection building in creating modern large test collections, IR researchers investigate on pooling methods that are more feasible in comparison to traditional pooling in term of the assessment cost. Moffat et al. [125] argued that the importance of all the pooled documents are not the same in building reusable test collections that are able to comparatively rank retrieval systems. They proposed considering of relevance likelihood of documents in creating the pools.

Cormack et al. [36] proposed a move to front pooling approach, which examines documents in order of their relevance likelihood among submissions. In fact, a submission that has more recently retrieved a relevant document is assumed to more likely retrieve another relevant document. Carterette et al. [25] did an experiment with an aim of judging the minimum number of documents essential to rank the pooled systems correctly. Specifically, they consider a document whose relevance has the highest effect in differentiating systems as the next document to judge.

Other work focuses on creating a more effective pool by using more diverse pooled runs. To this aim, relevance feedback is used to retrieve a new set of results in order to improve the pool effectiveness [93, 148]. Moreover, in order to build a reusable test collection, Carterette et al. [26] proposed an experimental design, which collects evidence for or against three types of reusability (i.e., within-team, between team and participant comparison) during collecting judgments.

As creating test collections for large dynamic collections using pooling is a very difficult problem, there are some researches focusing on simulation of judgments and test collection building phase. Soboroff et al. [149] did an extensive study on evaluating retrieval system and ranking them without using relevance judgments. Specifically, they randomly chose relevant documents from a pool of retrieved documents, and state

that they are able to distinct best performing runs from the worst runs according to the actual system ranking based on the official TREC qrel. Moreover, Carterette et al. [27] proposed dynamic test collections, in which they simulate users interaction with an aim of evaluating retrieval systems.

Rather than focusing on pooling itself, the current chapter focuses on the problem of how to update an existing test collection with sparse judgments, in case there are new documents with judgments available. Closest in spirit to our work is Soboroff [147], who studied how the GOV2 collection becomes outdated due to the changing Web, looking the effects of pages that disappear and change, and did experiments with simulated re-judging of changed pages. Soboroff also makes the suggestion to judge new pages not included in the original corpus, but doesn't do any experiments on this, and the current chapter addresses this head-on.

4.2.2 Reusability of Test Collections

There is quite some literature on the reusability of test collections. LOU is the standard test for evaluating reusability of test collections in ranking non-pooled systems. Leave-one-run-out (LORO) is a preliminary version of this test that introduced by Zobel [186] to identify effects of missing relevant documents in evaluating non-pooled systems. To be more specific, using traditional top-n document pooling has a disadvantage of identifying a fraction of relevant documents, which underestimates effectiveness of a technique that did not have an opportunity to contribute to the pool. To estimate this disadvantage of pooling that contributes negatively in reusability of test collections, Zobel [186] selected a run, created a pool using all runs, and then excluded all documents contributed uniquely by the selected run from the pool. They repeated the process for all the runs contributed in the pool.

In LORO, let r denote a run contributed by one of the participated teams during test collection building process using pooling, and let D_r denote a set of documents contributed to the pool by a run r. A set of unique documents contributed by run r is defined as $U_r = D_r - \cup_{r' \neq r} D_{r'}$. Let J denote the complete set of judged documents. Then, for each pooled run, the run is evaluated by $J - U_r$ in LORO test. Rank correlation of the runs ranking based on the LORO test collection and runs ranking based on the complete set of judged documents J is an indicator of a test collection reusability based on LORO test.

Since runs contributed by a same team are similar, leaving all contributions of a team out (i.e. leave-one-team-out (LOTO) [22, 160]) is another reusability test, which is more critical in case teams submitted several similar runs, thereby reducing the number of uniquely retrieved documents in individual runs. In LOTO, let t denote a team contributed to a pool, and let D_t denote a set of documents contributed to the pool by a team t. A set of unique documents contributed by team t is defined as $U_t = D_t - \cup_{t' \neq t} D_{t'}$. Then, for each participated team, runs contributed by the team is evaluated by $J - U_t$ in LOTO test. Rank correlation of the runs ranking based on the LOTO test collection and runs ranking based on the complete set of judged documents J is an indicator of a test collection reusability based on LOTO test. Sakai [143] proposed take-just-one-team and take-just-three-team experiments to identify effects of missing judgments on a number of evaluation metrics (e.g., AP and bpref). In take just one

team test, for each participated team, runs contributed by the team is evaluated by D_t. Rank correlation of the runs ranking based on the take-just-one-team test collection and runs ranking based on the complete set of judged documents J is an indicator of a test collection reusability based on the test. The difference between take-just-three-team and take-just-one-team tests is that take-just-three-team test uses $\cup_{t \in T} D_t$ rather than D_t for each team, in which T is a set of chosen teams from the set of teams contributed in a pool.

We will use the standard reusability tests such as LOU, which simulates that a pooled run becomes non-pooled, but also propose a counterpart that simulates pooling for non-pooled runs.

4.2.3 Test Collection Building and Reusability in TREC Contextual Suggestion Track

TREC contextual suggestion track provides a highly personalized and contextualized task based on dynamic collections. As it is detailed in [72, 75], personalization and dynamic nature of the track's data collection contributes negatively in reusability of the test collection created in TREC 2014. Based on these observations, in order to avoid creating test collections that are not reusable, the track organizers decided to create a fixed collection from open Web [42]. However, although a fixed collection has been used in 2015, the TREC 2015 contextual suggestion track test collection is not reusable [76].

As it is discussed in [76], it is clear that even by fixing the data collection in TREC 2015, the impact of personalization on the reusability of the TREC contextual suggestion track test collection is significant. In TREC 2016 contextual suggestion track, as detailed in the previous chapter, the track organizers proposed multi-depth pooling approach, in which they tried to minimize pooling bias [76]. As TREC 2015 and 2016 contextual suggestion tracks use a dynamic collection, our proposed approach can be used to maintain their test collection reusability in case of a decrease in their reusability due to the dynamic nature of the collection.

This chapter focuses on the unique, dual setup of the TREC 2014 track, and reports much of the internal analysis done over the years that motivated the choice to look for alternatives like the reranking task that was central to the track in later years. While the re-ranking setup is a pragmatic way to avoid some of the hard problems with (lack of) reusability, it does so at the cost of a considerable loss of experimental power. Hence the analysis of this chapter remains valuable, and provides insight into the underlying deep and fundamental problems of reusable test collection building.

4.3 Test Collection Reusability

This section studies the reusability of the test collection, aiming to answer our first research question: *How reusable is the OpenWeb and ClueWeb12 test collections of the TREC contextual suggestion?* Our main finding is that, according to the LOU test and the fraction of judged documents detailed in this section, both the Open Web and ClueWeb12 test collections have a low degree of reusability.

Table 4.1: TREC 2014 Contextual Suggestion test collection statistics

Subset	#context-profile	# Venues	Depth	avg # judged documents	#Runs	#Teams
Open Web	299	8,441	5	85	25	14
ClueWeb12	299	2,674	5	27	6	3

4.3.1 Experimental Data

In this chapter, we have used the unique setup of the TREC 2014 Contextual Suggestion track. This track allows participants to submit their venue recommendation runs' results based on either open web (in the form of a valid URL) or ClueWeb12 dataset (in the form of a valid ClueWeb12 document ID). In TREC 2014, 31 runs submitted by 17 teams (with 14 teams submitting 2 runs). Among these submissions, 6 runs belong to 3 out of 17 teams who made their submissions based on the ClueWeb12 dataset, and the rest are based on the open web.

In contextual suggestion, a topic consists of a pair of both a context (a North American city) and a profile (consisting the requester's likes and dislikes of venues in another city). For example, a requester's preferences and their ratings of attraction in Chicago, IL are used to recommend venues to visit in the new city of Buffalo, NY. Runs were pooled at depth 5 and in total 299 context-profile pairs, which has 112 judged documents in average, were judged. A short summary of the TREC 2014 contextual suggestion test collection is given in Table 4.1.

4.3.2 Leave Out Uniques Analysis

We first look at the question: *How reusable is the test collection for evaluating non-pooled systems?* Specifically, we perform both the leave-one-run-out [186] and leave-one-team-out [22] experiments to see what would have happened if a run had not contributed to the pool of judged documents.

In order to evaluate the test collection reusability in evaluating non-pooled systems, Kendall's τ, which is a standard metric in measuring system rankings correlation, is used. This metric is formulated as follows:

$$\tau = \frac{C - D}{N(N-1)/2},$$

where C is the number of concordant pairs, D is the number of discordant pairs, and N is the number of systems in the given two rankings [173]. In addition to the Kendall's tau that is not promising in some conditions [24, 35, 173], AP Correlation Coefficient is used to measure system rankings' correlation more precisely. AP Correlation is formulated as follows:

$$\tau_{AP} = \frac{2}{N-1} \cdot \sum_{i=2}^{n} \left(\frac{C(i)}{i-1}\right) - 1,$$

where $C(i)$ is the number of systems above rank i and correctly ranked [173]. Moreover,

average percentage difference of common IR metrics considering before and after LOU test is also measured that will show the effect of being pooled or not in systems' scores.

Leave One Run Out

In leave-one-run-out (i.e., LORO) experiment, for each pooled run, the run's unique judgments are excluded from the test collection and it is evaluated based on the new test collection in terms of P@5, MAP and bpref.[1] Our main aim in this experiment is finding the correlation of the system ranking in the case that they are pooled and judged in the test collection building process with the one ranked based on the assumption that the systems are not pooled.

As it is shown in Figure 4.1, leave-one-run-out system ranking's correlation with the actual system ranking for both open Web and ClueWeb12 test collections are lower than scores reported as reusable test collections in previous researches. Specifically, Kendall's τ of the LORO experiment based on the MAP metric are 0.66 and 0.46 for open Web and ClueWeb12 test collections, respectively, which are much lower than 0.9 that is the threshold usually considered as the correlation of two effectively equivalent rankings [159]. According to Figure 4.1, even rank correlation based on bpref metric, which works better than precision based metrics for evaluating systems based on incomplete test collections, is not acceptable. Moreover, difference between actual P@5, MAP and bpref and the ones based on LORO test indicates that scores of systems are considerably underestimated by excluding their unique judgments from the test collection.

Leave One Team Out

In order to study the reusability problem of the Contextual Suggestion test collection more precisely, we study a more realistic LOU experiments. According to the observation made in [72], open web contextual suggestion runs submitted by each team is based on a similar or a same data collection. Therefore, leave-one-team-out (i.e., LOTO) is a better indicator of the test collection reusability in evaluating a new non-pooled run, which might use a completely different collection than the ones used by the pooled runs. According to this experiment, leaving one team's judgments out has a dramatic effect on both Open Web and ClueWeb12 runs' evaluation. The effect is more considerable in ClueWeb12 test collection. Specifically, MAP score of 3 out of 6 ClueWeb12 runs is 0 after leaving their teams' judgments out of the test collection. In fact, P@5, MAP and bpref scores are dropped to zero or almost zero in LOTO test of ClueWeb12 test collection.

In Figure 4.2, result of LOTO test shows that, in some condition, the correlation of the LOTO system ranking with the actual system ranking based on the official test collection is higher than correlation measured in the LORO test. This higher rank correlation is made by the effect of the lack of judgments and low number of systems contributed in the ClueWeb12 pool. The effect of lack of judgment for the data collection

[1] The track uses P@5 as main measure, and also supplies MRR and a modified time-based gain (TBG) measure. As we are dealing with sparse judgments, we opt to include MAP which is know to be very stable, and bpref which is designed to be stable under incomplete judgments. Experiments (not reported) confirm that MRR is very unstable and that TBG resembles the P@5 results.

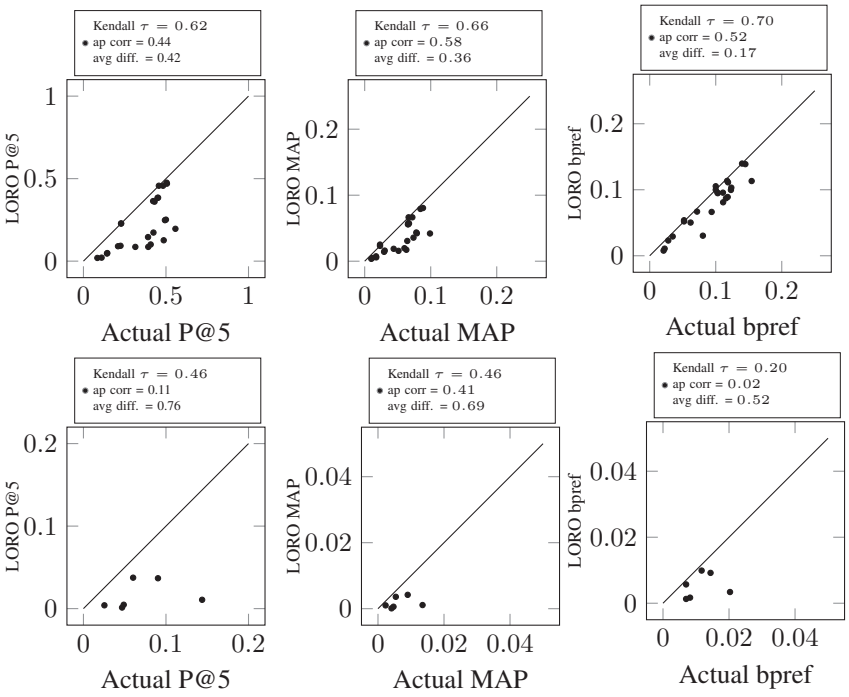


Figure 4.1: Difference in P@5, MAP, and bpref based on the leave one run out (LORO) test of OpenWeb runs (top) and ClueWeb12 runs (bottom).

is crystal clear by looking at the mean percentage difference in the LOTO test. It shows that the scores are dropped to zero or almost zero in LOTO test. Therefore, as the system ranking based on a tiny fraction of judged documents is not reliable, the higher LOTO score in comparison to the LORO score does not mean anything concrete and the number of judgments is insufficient to reach a conclusion.

According to Figures 4.1 and 4.2, the test collection has a low degree of reusability, and it should be used with extreme care. We will study the causes of this in the rest of the section, starting with the fraction of judgments in the runs.

4.3.3 Fraction of Judged Pages

We now look at the question: *What is the fraction of judged documents?* We want to find out if the Open Web or ClueWeb12 contextual suggestion test collection have enough judgments for venues suggested in ranks beyond the pooling depth. To this aim, we have analyzed overlap@N [72] as the fraction of the top-N suggestions that is judged for the given set of topics:

$$Overlap@N(\langle C,P\rangle) = \frac{1}{|\langle C,P\rangle|} \sum_{\langle c,p\rangle \in \langle C,P\rangle} \frac{\#Judged@N(\langle c,p\rangle)}{N},$$

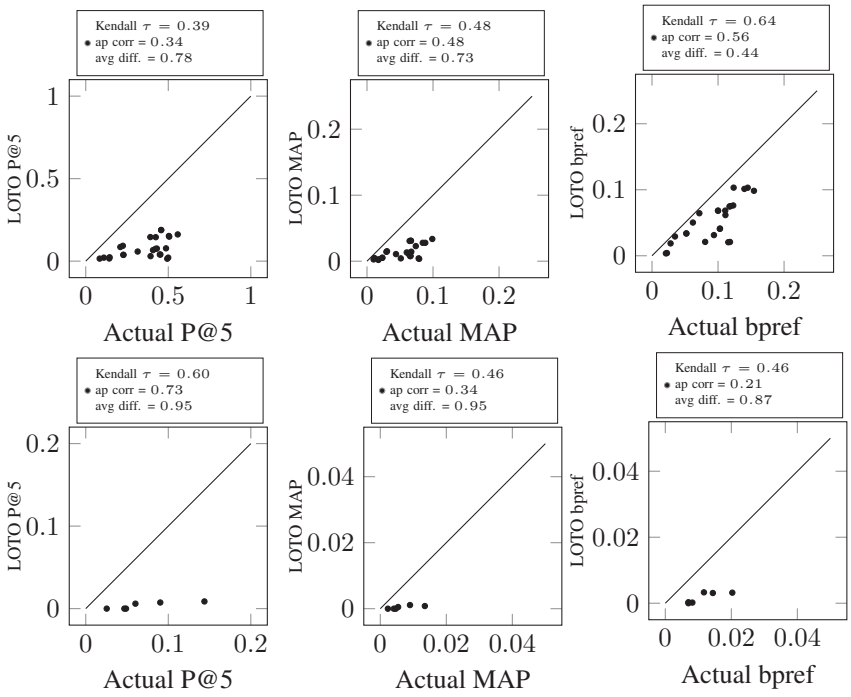


Figure 4.2: Difference in P@5, MAP, and bpref based on the leave one team out (LOTO) test of OpenWeb runs (top) and ClueWeb12 runs (bottom).

where $\#Judged@N(\langle c, p\rangle)$ corresponds to the count of judged suggestions for the given context and profile pair $\langle c, p\rangle$ in the top-N suggestions, and $\langle C, P\rangle$ is a set of judged context and profile pairs. According to Figure 4.3, the personalized test collection overlap is dropped significantly after pool cut-off. This observation indicates that the test collection is incomplete in terms of recall and consequently, the pooled runs overlap is relatively low. Overlap@N in ranks deeper than pool cut-off is biased to the top-5 judgments, and as it is shown in Figure 4.3, overlap at rank intervals deeper than pool cut-off dropped even more than overlap@N. In particular, overlap at rank intervals is almost zero after pool cut-off, which shows how serious is the lack of judgments in ranks deeper than the pool cut-off.

4.3.4 Impact of Personalization and Pool Depth

In this part, we answer the question: *What is the impact of personalization on the fraction of judged documents?* Same as the TREC Contextual Suggestion open test collection [72], personalization and the shallow pool depth affect the ClueWeb12 personalized test collection's reusability. As it is discussed above, the personalized contextual suggestion test collection is not reusable and it should be used with extreme care. However, in order to build a reusable test collection for the venue recommendation off-line testing, we first depersonalize the official test collection to see whether the non-personalized test collection has enough judgments for reliably ranking systems. We have used the

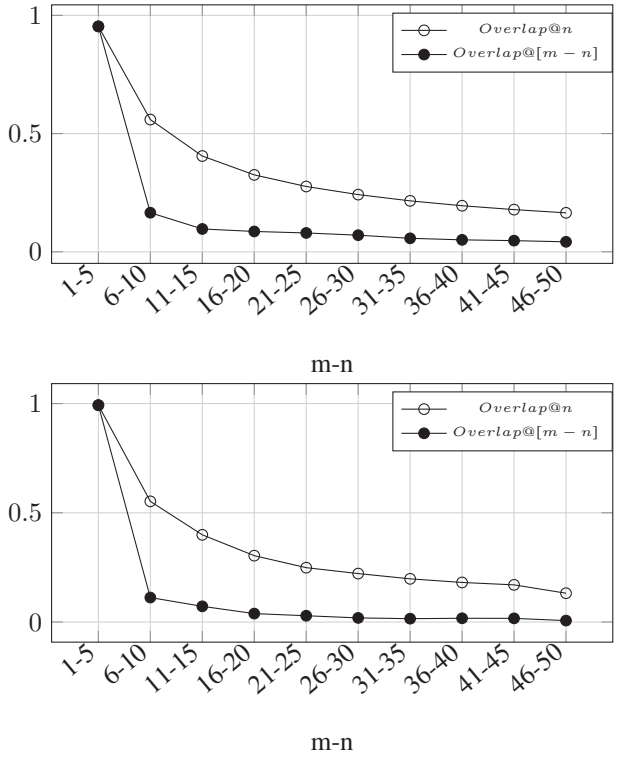


Figure 4.3: Overlap@n and Overlap@[m-n] over rank intervals in Open Web (top) and ClueWeb12 (bottom) test collections. In each rank interval, m is representative of highest rank in the interval and n is representative of lowest rank in the interval.

Borda count fusion over profiles to build non-personalized runs based on the pooled personalized runs. For the evaluation purpose, any suggestion, which judged as a relevant suggestion for the given city and one of the judged profiles, is counted as relevant suggestion for the given city.

Figure 4.4 demonstrates that personalization has a considerable effect on the overlap of the ClueWeb12 test collection, which is mainly for its effect on the pool depth. We have observed a similar result using Open Web test collection [72]. In order to solve reusability problem of the test collection that is affected by the shallow pool depth, we propose a novel approach to improve number of judged documents for each city and profile pair, which would have a positive effect on test collections reusability.

To summarize, in this section we investigated the reusablity of the TREC contextual suggestion track's OpenWeb-based and ClueWeb12-based test collection. The outcome is rather negative: the system rank correlation in the LOU test is below the reusable test collections threshold, with MAP and bpref scores close to zero; the fraction of judged documents after the pooling depth plummets down; and the combination of shallow pools over personalized runs aggravates the problem considerably. One can debate whether a 90% system rank correlation is a realistic goal for personalized test

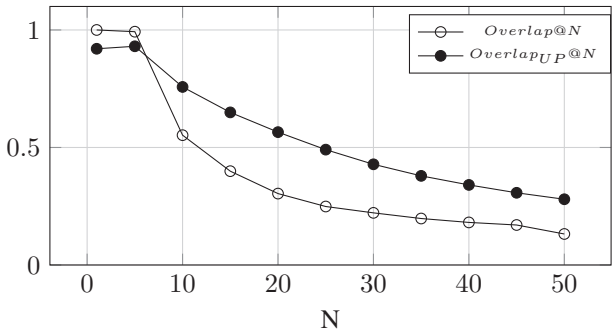


Figure 4.4: Effect of personalization on $Overlap@N$. $Overlap_{UP}@N$ is $overlap@N$ based on the non-personalized test collection

collections, judged to a limited pool depth of 5, in comparison to the traditional ad hoc search settings. As personalization substantially lowers the fraction of judged pages, which need not only be topically relevant but also fitting the user's profile, we certainly require more stable rankings than observed for the ClueWeb12 test collection. To address this, we will propose a way to expand or update an existing test collection in the next section.

4.4 Expanding Test Collections

In this section, we detail how we expand a test collection in order to improve its reusability. This approach holds the potential to increase the reusability of a test collection in scenarios such as dynamic domains where documents content changes and a new (relevant) documents appear [147].

4.4.1 Injecting Judged Documents

Our approach is rather straightforward: in case a fixed test collection becomes outdated and systems return documents not included in the outdated corpus, we simply judge the new documents, and merge them into an expanded test collection. We metaphorically hide the new documents in the old collection as Easter eggs for systems to retrieve as in an Easter egg hunt.

So assume we have a test collection based on a fixed corpus, which is not reusable. This test collection is formulated as follows:

$$TC_f = \{(t, d, r) | t : T, d : D_f, r : R_f\},$$

where t is a topic from the judged topics set (i.e., T), d is a document belongs to the fixed corpus, and r is a relevance judgment from judgments given for the fixed corpus (i.e., R_f). Moreover, consider that we have a set of new pages for the same problem and a same topic set, of which some or all are judged. This second set of judged documents has a similar formulation:

$$TC_s = \{(t, d, r) | t : T, d : D_s, r : R_s\},$$

where D_s is a set of documents from the secondary collection and R_s is a set of judgment for some documents of the second corpus (which could be an open collection like the web).

In order to use the second test collection for expanding the test collection, for each document $d_1 \in D_s$, the document is injected to the fixed collection (i.e., D_f), and relevance judgments of document d_1 (i.e., $\{(t, d, r)|t : T, d == d_1, r : R_s\}$) are added to the fixed test collection (i.e., D_f). Finally, each judgment in the new test collection is an instance of the following set:

$$TC_e = \{(t, d, r)|t : T, d : D_f \cup D_s, r : R_f \cup R_s\},$$

where d is a document judged in either the fixed test collection (i.e., D_f) or the secondary test collection (i.e., D_s), and r is a relevance judgment based on either the relevance judgments set created for the fixed collection (i.e., R_f) or the secondary relevance judgments set (i.e., R_s).

4.4.2 Expanded Contextual Suggestion Test Collection

The unique setup of TREC Contextual Suggestion track, which is discussed in Section 4.3, allows us to test our approach on this test collection. To this aim, we inject the judged open contextual suggestions into a fixed contextual suggestion collection (i.e., ClueWeb12 touristic sub collection, which is provided by the TREC organizers). To be specific, the ClueWeb12 sub collection contains 176,970 documents focusing on the touristic domain, and there are 7,434 judged open web documents as candidates to be merged into this collection.

The expansion of the test collection consists of two steps:

- First, we determine which open web pages are also included in ClueWeb12, based on the mapping of [67]. We retain the copy of the page in ClueWeb12, as these pages tend to describe venues and still describe the same entity, although an alternative is crawl the pages and update them. The qrels are expanded with the judgments for this page.

- Second, for remaining open web pages, we have either fetched rest of the web pages from the web or used the touristic aggregators' websites' (e.g., Yelp) API to gather the judged web pages' textual content. These judged documents are added to the collection, and the qrels are expanded with the judgments for this page.

The new qrels are substantially richer. To be specific, the contextual suggestion ClueWeb12 test collection has 8,043 judgments including 682 relevant judgments, and we add 25,407 open web judgments including 9,738 relevant judgments into that.

To summarize, in this section we investigated an approach to update or expand a test collection with a secondary set of judged pages, aiming to increase the reusability of the resulting test collection. The general approach is to simply "hide" the judged pages in the original collection, with the goal of systems to retrieve the relevant pages amongst the rest of the collection. The above scenario is a common case in all dynamic

Table 4.2: Personalized non-pooled runs and their descriptions. In these runs, personalization is done based on users' positive profiles.

Ranker	Description
LM JM BQ	Language modeling, default JM smoothing (i.e., $\lambda = 0.4$), Boolean personalization
LM JM	Language modeling, default JM smoothing (i.e., $\lambda = 0.4$)
LM two-stage	Language modeling, default two-stage smoothing (i.e., $\mu = 2{,}500$ and $\lambda = 0.4$)
LM JM2	Language modeling, JM smoothing and $\lambda = 0.001$
LM Dir.	Language modeling, default Dirichlet smoothing (i.e., $\mu = 2{,}500$)
Okapi	Okapi, default parameters (i.e., $k_1 = 1.2$, $b = 0.75$ and $k_3 = 7$)
tfidf	tf.idf, default parameters (i.e., $k_1 = 1.2$ and $b = 0.75$)
Okapi2	Okapi, $k_1 = 0.001$, $b = 0.001$ and $k_3 = 0.001$
tfidf2	tf.idf, $k_1 = 0.001$ and $b = 0.001$

domains, such as online services on the web. We applied the approach to the case of the TREC contextual suggestion track, merging the large set of judged open web pages into the ClueWeb12 based collection, leading to an updated test collection with a far greater number of judged documents. In light of the low degree of reusability of the ClueWeb12 test collection, as discussed in Section 4.3, the hope is that the expanded test collection will have a higher degree of reusability, which we will investigate in the next section.

4.5 Reusability of the Expanded Test Collection

In this section, we look at the question: *How reusable is the expanded test collection containing judged open web documents?* Our main finding is that the experimental results show that the expanded test collection is reusable based on the LOU test.

4.5.1 Leave Out Uniques

We evaluate reusability of the test collection by discussing the correctness of the non-pooled system ranking based on the expanded test collection. Specifically, we look at the following research question: *How reusable is the expanded test collection for ranking systems?*

To test reusability of the expanded test collection, we build nine different personalized non-pooled contextual suggestion runs using language modeling with different smoothing approaches, okapi and tf.idf retrieval models. All these models built based on Indri IR tool. We intentionally pick optimal and suboptimal parameter settings so that we have a realistic variation in retrieval effectiveness. Our main goal of building these non-pooled runs is to investigate whether the expanded test collection is able to discriminate high quality non-pooled runs from the low quality ones or not. A short summary of these runs is given in Table 4.2. We also build nine non-personalized non-pooled runs based on the Indri, which is going to be used in evaluating the non-personalized

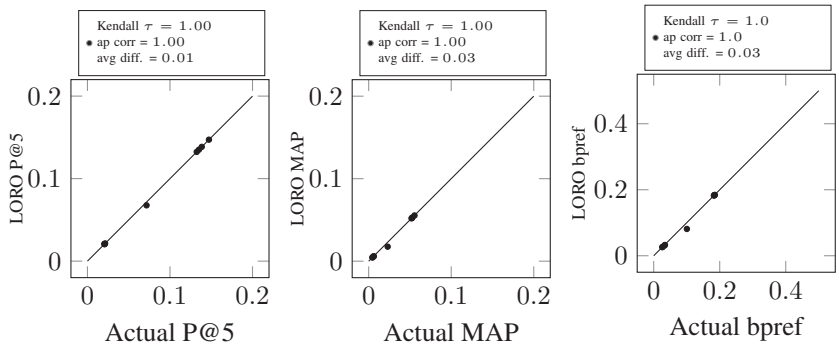


Figure 4.5: Difference in P@5, MAP, and bpref based on the leave one run out (LORO) test on the expanded test collection.

expanded test collection reusability. These runs are similar to the personalized runs, and use the context or city in combination with a generic set of touristic categories used for all the profiles.

In this experiment, we would like to test whether the expanded test collection is effective enough in ranking high quality runs higher than the low quality ones or not. To this aim, two groups of personalized runs are built to retrieve suggestions relevant to the given city name and profile. One of them is a group of runs based on personalized query expansion using a group of defined touristic categories (i.e., LM JM BQ, LM JM, LM two-stage, LM JM2 and LM Dir.). The other one is based on retrieving relevant suggestions to the given city name, and then ranking suggestions based on similarity of suggestions to the given profile (i.e., Okapi, tfidf, Okapi2 and tfidf2). We know that the second group of runs might miss some suggestions relevant to the given profile that the city name is not mentioned explicitly in their contents. For example, some of the relevant suggestions might include name of a city close to the given city name rather than the city name. Moreover, in the second group, rather than using language modeling, less effective retrieval models like tfidf and okapi are used. We expect lower rank for the second group of runs in comparison to the first more effective runs.

As it is shown in Table 4.4, the expanded test collection is able to discriminate these two groups of runs, and also rank relatively reasonable within each group of runs. On the other hand, Table 4.3 indicates system ranking of the same runs based on the official TREC test collection, which shows that the official test collection is not able to rank systems in a logical order. In order to test reusability of the test collection, LOU test is done using LORO test. According to Figure 4.5, the actual system ranking is exactly same as the LORO system ranking, and they have the highest rank correlation in terms of Kendall's τ and AP correlation. Specifically, Kendall's τ and AP correlation of this test is 1, which presents the strongest possible evidence for the reusability of the expanded test collection for ranking non-pooled personalized systems.

However, some of these non-pooled runs retrieve similar set of suggestions, due to the fact that all of these runs are based on a same index and same personalization approach. This effect the outcome of the LORO test, which may be too optimistic, and it is not possible to do the more critical LOTO test in this particular setup. The effect of

Table 4.3: Personalized non-pooled system ranking based on MAP using official test collection and their overlap

Run	P@5	MAP (%)	bpref	Overlap@50 (%)
LM Dir.	2.94	0.41	2.15	11.49
okapi	1.87	0.26	1.61	14.39
tfidf	1.74	0.26	1.59	14.30
okapi2	2.01	0.24	1.50	13.97
tfidf2	2.01	0.24	1.46	13.78
LM JM2	0.40	0.07	0.98	3.81
LM JM BQ	0.33	0.06	0.83	3.33
LM JM	0.40	0.06	0.83	3.31
LM two-stage	0.40	0.06	0.80	3.19

Table 4.4: Personalized non-pooled system ranking based on MAP using expanded test collection and their overlap

Run	P@5	MAP (%)	bpref	Overlap@50 (%)
LM JM BQ	14.72	05.55	18.49	31.57
LM JM	13.85	05.29	18.35	31.45
LM two-stage	13.51	05.25	18.44	31.49
LM JM2	13.24	05.19	18.23	31.43
LM Dir.	7.16	2.30	10.05	27.28
okapi	2.14	0.62	3.41	17.69
tfidf	2.07	0.58	3.24	17.44
okapi2	2.07	0.46	2.71	16.28
tfidf2	2.07	0.46	2.59	16.08

the runs' similarity on doing LORO experiment motivates us to run another experiment to test the effectiveness of the expanded test collection in ranking new systems in the next section. But first we investigate whether the runs do indeed retrieve the judged documents we added to the expanded test collection.

4.5.2 Retrieving Judged Documents

In order to evaluate effectiveness of the expanded test collection, we study the research question: *Are retrieval models able to retrieve the judged open web documents?* Recall that in this section we use a new set of runs on the expanded test collection, based on the touristic subset of ClueWeb and the Open Web runs, making these results not directly comparable to those in Section 4.3.

Figure 4.6 shows overlap@N of the non-pooled runs with expanded test collection as well as the official contextual suggestion test collection. This experiment indicates that the injecting judged documents approach has a considerable impact on the personalized

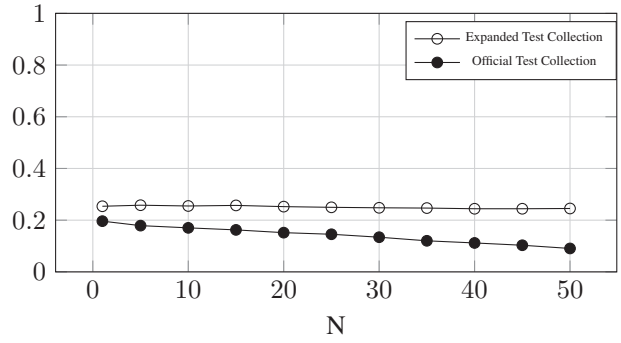


Figure 4.6: Overlap@N of non-pooled runs: official test collection versus expanded test collection for personalized runs.

Table 4.5: Non-personalized non-pooled system ranking based on MAP using official test collection and their overlap

Run	P@5	MAP (%)	bpref	Overlap@50 (%)
LM Dir.	11.49	0.79	2.99	30.08
okapi	4.68	0.47	2.41	30.68
tfidf	3.83	0.44	2.38	30.38
okapi2	4.68	0.39	2.12	28.72
tfidf2	5.11	0.38	2.17	29.14
LM JM BQ	3.83	0.18	1.20	9.74
LM JM	3.83	0.18	1.23	9.87
LM JM2	3.83	0.18	1.26	10.04
LM two-stage	3.83	0.17	1.19	9.65

test collection fraction of judgments. In particular, overlap@50 is improved from 0.14 to 0.26, which is 85% improvement in the fraction of judgments.

As discussed in Section 4.3, depersonalization of the contextual suggestion has a great impact on the fraction of judged documents. Therefore, effectiveness of non-personalized non-pooled runs in hunting injected judged documents in the ClueWeb12 tourist sub collection is also studied. Table 4.6 indicates that injecting judged documents in a fixed corpus has a great impact on the non-personalized test collection fraction of judgments. In addition, same as personalized expanded test collection, the system ranking based on the non-personalized expanded test collection is reasonable. However, according to Table 4.5, system ranking of the same runs based on the official TREC test collection shows that the official test collection overlap is poor and it is not able to rank non-personalized systems in a logical order.

To summarize, in this section we investigated the reusability of the expanded contextual suggestion test collection. The result is positive: we determined the reusability by doing a LOU analysis, leading to perfect system rank agreement over a set of nine systems. While all these systems did not contribute to the pool, the stability of the

Table 4.6: Non-personalized non-pooled system ranking based on MAP using expanded test collection and their overlap

Run	P@5	MAP (%)	bpref	Overlap@50 (%)
LM JM BQ	48.94	15.30	22.91	87.10
LM two-stage	50.21	15.27	22.85	87.14
LM JM	49.36	15.21	22.84	87.14
LM JM2	49.36	15.14	22.81	87.19
LM Dir.	26.81	05.82	12.74	66.38
okapi	4.68	0.91	3.87	36.42
tfidf	3.83	0.86	3.80	36.29
okapi2	4.68	0.71	3.30	33.36
tfidf2	5.11	0.70	3.35	33.95

ranking is a reassuring outcome. In order to explain the ranking stability we looked at whether systems are indeed retrieving the inserted judged pages, and found that a fair and stable fraction of judged documents is retrieved, more than doubling the fraction of judged documents, and that this fraction is gently decreasing of the ranking. The effect of personalization remains large, and de-personalized versions of the qrels ignoring the profile lead to substantially higher fractions of retrieved judged documents. This gives strong support to the test collection expansion approach proposed in this chapter. The positive results hold for the system rank comparison among a set of non-pooled runs, and with shallow pools we may expect a substantial pooling effect when comparing pooled and non-pooled runs, which we will investigate in the next section.

4.6 Impact of Simulated Pooling on the Reusability

In this section, we answer the question: *What is the impact of simulated pooling on the reusability of the expanded test collection?* We propose a counterpart to the LOU test that simulates the impact of pooling in a Leave In Uniques test. Our main contribution is that this approach is a stricter and more powerful reusability test, that may uncover risks to a fair comparison of pooled and non-pooled runs even prior to judgments being available.

4.6.1 Simulated pool and its impact on the reusability

As all the nine runs used in evaluating reusability of the expanded test collections are non-pooled runs, testing reusability of the expanded test collection based on LOU test does not lead to a definite conclusion of the test collection reusability. In fact, the LOU test is designed to test reusability of test collections based on pooled runs. Therefore, the reusability of the expanded test collection remained unanswered in previous sections.

In addition to *LOU* test that is discussed in last Section, in this part, we answer the question: *Does non-pooled system ranking change by adding more simulated judgments?* In fact, we would like to analyze whether the test collection is stable in

ranking non-pooled systems or it might change by adding simulated judgments.

In this experiment, for each non-pooled run, we artificially simulate judgments of unjudged documents based on the same distribution of relevant documents in that specific run. To this aim, a weighted random variable, whose weight is the fraction of relevant documents among the judged documents up to the rank of the given unjudged document, is used to simulate the judgment. If relevance judgment of a document is available in our test collection, we do not use a simulated judgment for the document. If relevance judgment of a document is not available in our test collection, we use simulated judgment as it is discussed above. In this case, the final simulated judgment of a document is selected based on majority votes of the 9 runs have been used in this experiment. In the cases that a document gets equal number of relevant and irrelevant votes, the final simulated judgment is considered as relevant. Using simulated judgments, we build 50 different test collections, which in addition to judgments of the original test collection for the judged documents have the simulated judgments for the unjudged ones with different simulated pool depth from 1 to 50.

In Figure 4.7, Kendall's τ_{sig}, which only consider significant inversions [35], of the system ranking based on the expanded test collection with the one based on simulated ones having simulated judgments up to depth N is measured. In this experiment, we use a paired Student's t-test with $\alpha = 0.05$ to find significant inversions (i.e., $p < \alpha$). As it is shown in this Figure, the rank correlation has a lot of rise and fall up to pool depth 33, but it is much more stable having deeper pools from depth 34 to 50.

This experiment indicates that the expanded test collection is more similar to the simulated test collections having deep pools rather than test collections having shallow simulated pools. We had a similar observation based on overlap@N metric, which demonstrates that the expanded test collection has high overlaps in all of the pool depths. This is a good signal that shows using this expanded test collection let us evaluate systems based on IR metrics at deep ranks.

Moreover, according to the rank correlation of the system rankings based on the expanded test collection and the most complete simulated test collection, the expanded test collection is stable and can be used as a reusable test collection for offline testing. Specifically, τ_{sig} of the system ranking based on the MAP metric measured by the expanded test collection with the system ranking based on the complete simulated judged test collection is 1.0, which is considered as the correlation of two effectively equivalent rankings [159]. We make the same observation based on Kendall's τ, but with lower rank correlation scores due to inversions in case of very small differences (not presented in details here).

4.6.2 Simulated test collection pool cut-off

In this part, we look at the question: *What is the most effective pool cut-off based on the simulated test collection?*

In order to study the effective pool depth of the simulated pools, we study the rank correlation of the system ranking based on the deepest simulated pool (with 50 pool depth) with system rankings based on different simulated pool depths. Also according to Figure 4.7, the system ranking based on the test collection with pool depth beyond rank 35 is quite similar to the one evaluated by complete pool depth, and adding more

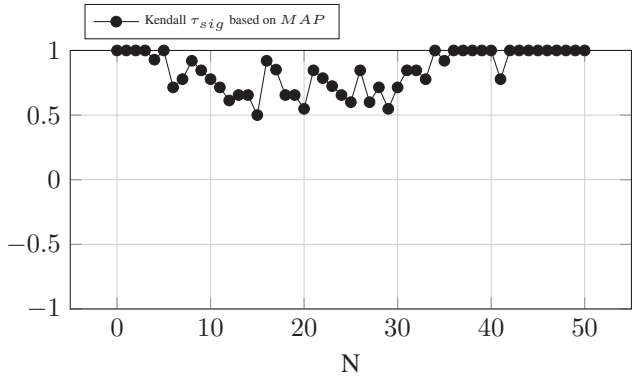


Figure 4.7: *Kendall* τ_{sig} of system ranking based on the expanded test collection (as a ground truth) and simulated test collection having different pool depth.

judgments by increasing pool depth does not have significant effect on system rankings based on MAP.

4.6.3 Simulated Pooling Effect and Leave Uniques In Test

We now look at the following research question: *How to estimate simulated pooling bias of the test collection in ranking non-pooled runs?*

As we created a new expanded test collection, none of the runs on the expanded corpus was pooled, and none of the pooled open web runs retrieves more than a fraction of pages in the corpus. Hence, we propose a variant of the LOU test that simulates the pooling effect.

In the previous experiments, we observe the test collection is reusable for ranking non-pooled systems, but the test collection effectiveness in ranking pooled systems is questionable. To this aim, we propose Leave-One-Run/Team-In test for evaluating test collections effectiveness in ranking pooled systems in the case that the test collection is not built based on pooling approach.

In Leave-One-Team-In (LOTI) test, in principle, for each team has not contributed in the pool, all the unjudged documents of the team's non-pooled runs have to be judged up to a given pool depth and then added to the qrel. The team's non-pooled runs are evaluated based on the LOTI qrel created for the team. This test is done for all the teams not contributed in the pool. The system ranking correlation between the system ranking based on the original qrel and the system ranking based on the LOTI test indicates effects of pooling in the system ranking using the test collection. Specifically, if the system ranking correlation is lower than a defined threshold, the test collection is not reusable.

In Leave-One-Run-In (LORI) test, for each non-pooled run, all the unjudged documents of the non-pooled run have to be judged up to a given pool depth and then added to the qrel. Then, the non-pooled run is evaluated based on the LORI qrel. This test is done for all the non-pooled runs. The system ranking correlation between the system ranking based on the original qrel and the system ranking based on the LORI qrel indicates effects of pooling in the system ranking using the test collection. Same as

LOTI test, if the system ranking correlation is lower than a defined threshold, the test collection is not reusable based on the LORI reusability test.

In this study, as we do not have different teams' runs that retrieve and rank documents based on the created extended corpus, we are not able to do the LOTI test. However, as it is detailed in Table 4.2, we have 9 different runs created based on the extended corpus as a data collection. Using these 9 non-pooled runs, we have done the LORI test using simulated judgments.

In this experiment, for each non-pooled run, the same distribution of relevant documents in that specific run is used to artificially simulate judgments of unjudged documents. To this aim, we simulate the judgment for each topic based on a weighted random variable, whose weight is the fraction of relevant documents among the judged documents up to the rank of the given unjudged document. For each run, we judge their unjudged documents and make 50 different simulated pools having pool depth from 1 to 50. The simulated judgments of the unique contribution of each run in the simulated pool, is used to create the LORI qrel. The simulated judgments of documents retrieved by multiple runs up to the simulated pool cut-off, is estimated by a weighted random variable having the following weight:

$$w = \frac{\#\ relevant\ simulated\ judgment\ of\ document\ d}{\#\ runs\ retrieved\ the\ document\ d\ up\ to\ the\ pool\ depth}$$

In this way, final simulated judgments of all the non-unique documents retrieved by multiple runs are estimated and then added to the LORI qrel.

To summarize, in this test, we leave simulated unique judgments up to pool depth of each simulated pooled run in the expanded test collection (i.e., the one that does not include simulated judgments), and evaluate them based on the new test collection. This is done for all the non-pooled runs, and the rank correlation of the non-pooled system ranking (as a ground truth) and the simulated pooled system ranking is an indication of the test collection reusability in ranking pooled and non-pooled systems. The proposed leave uniques in test is very practical to evaluate reusability of expanded test collections being maintained or test collections not built based on pooling. Moreover, this test can be used to evaluate reusability of test collections while building them using TREC-style pooling. In fact, at each point of time in test collection building, leave uniques in test can indicates whether pooling deeper will improve reusability or the test collection is effective enough in ranking retrieval systems. More investigation on this line of research based on leave uniques in test remains as a future work.

4.7 Reusability Based on Leave One Run In Test

In this section, we investigate on reusability of the expanded test collection using LORI test in order to answer the research question: *How reusable is the expanded test collection for ranking pooled systems?* Our main finding is that on this critical test, the expanded test collection is shown to be reusable for ranking personalized non-pooled and pooled runs based on the stable MAP and bpref metrics, but should be used with care on more unstable P@5 metric.

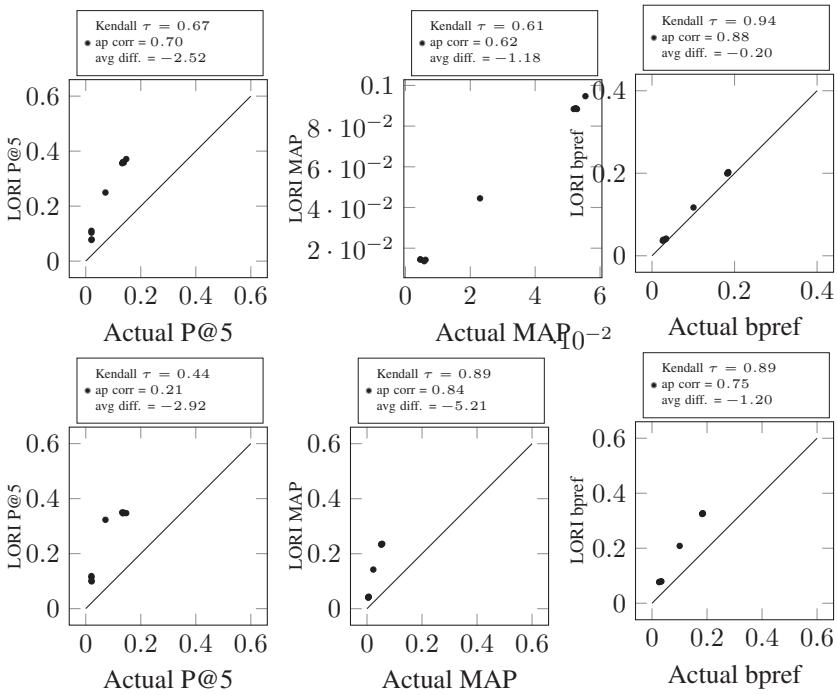


Figure 4.8: Difference in P@5, MAP, and bpref based on the leave one run in (LORI) test with pool depth 5 (top) and pool depth 50 (bottom) on the expanded test collection.

4.7.1 Reusability of Personalized Runs

In this part, we answer the following research question: *How reusable is the personalized expanded test collection for ranking simulated pooled systems?* Fraction of judged documents and the effectiveness of simulating judgments for the unjudged documents plays an important role in the LORI test. In particular, in this experiment, non-pooled runs have 25% judged documents in average and about 75% of the judgments in LORI experiment is simulated. Therefore, using a more effective approach for simulating judgments might change the rank correlation scores, and using 0.9 as a heuristic threshold of two equivalent rankings in testing effectiveness of test collections in LORI experiment is perhaps overly strict, so we will not treat it as a dichotomous cut-off but rather as a soft target.

Figure 4.8 shows the rank correlation metrics and score difference of runs in LORI test with simulated pool depth equal to 5 and 50. According to these figures, same as incomplete test collections, bpref is more stable and less overestimated in comparison to MAP and P@5. Specifically, Kendall's τ based on bpref in the LORI test having pool depth 5 is 0.94, which suggests that the expanded test collection is reusable for ranking personalized runs based on bpref metric. We have had a similar conclusion based on the LORI test with 50 as the pool depth.

As can be expected, the deeper simulated pools are more stable for system ranking based on MAP metric. In contrast to the Kendall's τ based on MAP in the LORI test

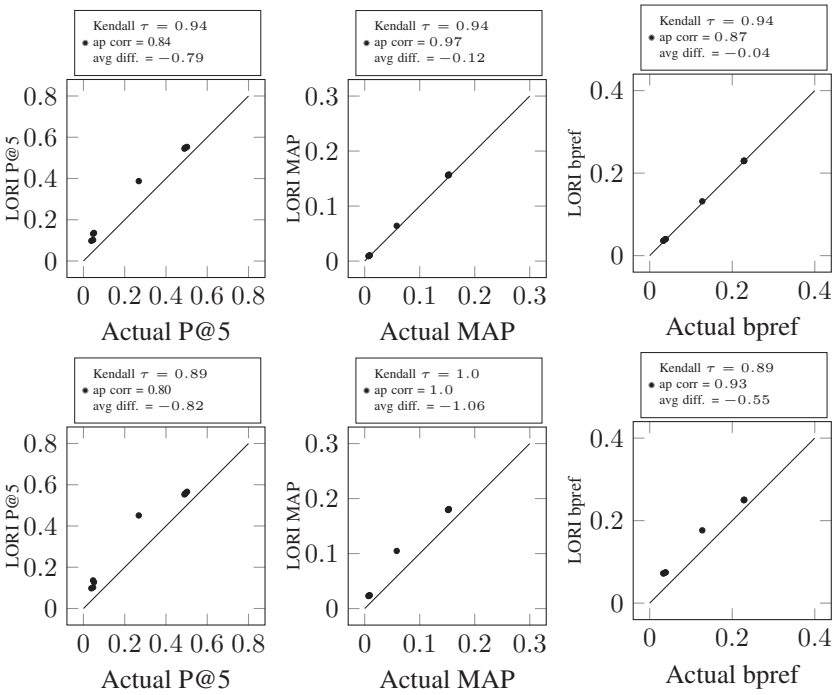


Figure 4.9: Difference in P@5, MAP, and bpref based on the leave one run in (LORI) test with pool depth 5 (top) and pool depth 50 (bottom) for the non-personalized expanded test collection.

with 5 as the pool depth, which is 0.61, the Kendall's τ based on MAP in the LORI test having pool depth 50 is 0.89. The experiment shows that if a test collection created based on the pool depth 5 be used for ranking personalized runs based on MAP metric, the system ranking would not be same as the system ranking based on MAP using the expanded test collection. On the other hand, if a test collection created based on the pool depth 50 be used for ranking personalized runs based on MAP metric, the system ranking would be very similar to the system ranking based on MAP using the expanded test collection. The Kendall's τ = 0.89 is just 1 percent less than the threshold usually considered as two effectively equivalent rankings in LOU test. Factoring in the noise due to the simulations being done in the LORI test, we consider this to suggest that the expanded test collection is reusable for ranking personalized systems based on MAP metric.

Unlike the earlier LORO test, which looked favorable at all measures, the outcome on the LORI test is mixed. Although the expanded test collection seems reusable based on MAP and bpref metrics, it suggests a low degree of reusability for ranking personalized runs based on P@5 metric. Specifically, the Kendall's τ based on P@5 in the LORI test with 5 and 50 as the pool depths are 0.67 and 0.44, respectively, which is far less than the heuristic 0.9 threshold usually considered as two effectively equivalent rankings in LOU test. This result suggests that the comparison of pooled and non-pooled systems should be done with care based on early precision metrics like P@5, as this measure is biased toward the pooled runs. This call to caution is not unexpected, as the early precision measure is known to be less stable and the test collection is build under challenging conditions, in particular the relatively shallow pools due to personalization over profiles.

4.7.2 Impact of Personalization and Pool Depth

In this part, we investigate on the following research questions: *What is the impact of personalization and pool depth on the reusability of the test collection?* According to the last experiment, low fraction of judged documents in personalized runs affects on the LORI test based on the P@5 as an early precision metric. Therefore, reusability of the non-personalized expanded test collection, which has a high overlap with the non-pooled runs, in ranking pooled systems is evaluated.

Figure 4.9 shows the evaluation of non-personalized expanded test collection based on LORI test with simulated pool depth equal to 5 and 50. According to these figures, the non-personalized test collection is strongly reusable in ranking pooled systems based on P@5, MAP and bpref metrics. Specifically, Kendall's τ of the LORI experiment having pool depth 5 is 0.94 based on all the tested metrics, which is higher than 0.9, the threshold usually considered as the correlation of two effectively equivalent rankings. We have got a similar result for the LORI test having pool depth 50. This means that the non-personalized expanded test collection is strongly reusable for ranking non-personalized pooled systems based on both early precision metrics and more stable metrics in incomplete test collections.

To summarize, in this section we investigated the reusability of the expanded test collection for ranking pooled systems based on the proposed variant of the LOU test in order to simulate the pooling effect of a set of non-pooled runs. This critical test

suggests that the expanded test collection is reusable for ranking personalized non-pooled and pooled runs based on MAP and bpref metrics, but not based on P@5 metric. We investigated the impact of personalization on the reusability, and found that the non-personalized test collection has a high degree of reusability based on all metrics including P@5, highlighting the challenges of test collections that use personalization and shallow pooling.

4.8 Discussion and Conclusions

In this chapter, we investigated the challenges of expanding or updating a test collection in a dynamic domain. We experimented with a novel approach to reusable test collection building, where we inject judged pages into an existing corpus, and have systems retrieve pages from the extended corpus with the aim to create a reusable test collection. In a way, we metaphorically hide the Easter eggs for systems to retrieve. The approach was motivated by, and applied to, the TREC Contextual Suggestion Track offering a personalized venue recommendation task, which allowed both submissions from a fixed corpus (ClueWeb12) as well as from the open web.

Our main research question was: *Can we build a reusable test collection for a dynamic domain by injecting judged documents into a test collection with sparse judgments?* Specifically, we answer following research questions: Our first research question was: *How reusable is the OpenWeb and ClueWeb12 test collections of the TREC contextual suggestion?* The outcome is rather negative: the system rank correlation in the LOU test is below 50%, with MAP and bpref scores close to zero; the fraction of judged documents after the pooling depth plummets down; and the combination of shallow pools over personalized runs aggravates the problem considerably. Our second research question was: *How to expand a test collection in order to improve its reusability?* Our approach is to simply "hide" the judged pages in the original collection, with the goal of systems to retrieve the relevant pages amongst the rest of the collection. The above scenario is a common case in all dynamic domains, such as online services on the web. We applied it to the case of the TREC contextual suggestion track, merging the large set of judged open web pages into the ClueWeb12 based collection, leading to an updated test collection with a far greater number of judged documents. Our third research question was: *How reusable is the expanded test collection containing judged open web documents?* The result is positive: we determined the reusability by doing a LOU analysis, leading to perfect system rank agreement over a set of nine systems. We found that a fair and stable fraction of judged documents is retrieved, more than doubling the fraction of judged documents, and that this fraction is gently decreasing over the ranking. The effect of personalization remains large, and de-personalized versions of the qrels ignoring the profile lead to substantially higher fractions of retrieved judged documents. Our fourth research question was: *What is the impact of simulated pooling on the reusability of the expanded test collection?* We proposed a variant of the LOU test in order to simulate the pooling effect of a set of non-pooled runs. Our fifth and final research question was: *How reusable is the expanded test collection for ranking pooled systems?* This critical test indicates that the expanded test collection is reusable for ranking personalized runs based on the stable MAP and bpref metrics. However,

the expanded test collection is not reusable for ranking personalized runs based on P@5 metric, which is an early precision based metric known to be less stable. Our investigation on the impact of personalization on the reusability shows that the non-personalized test collection has a high degree of reusability for all the metrics including the P@5 as the early precision metric, highlighting the challenges of test collections that use personalization and shallow pooling.

The leave uniques in reusability test is of independent interest, as it adds a new and critical test for analysing the reusability of the test collections we create and (re)use as official benchmarks deciding on the superiority of technical advances to systems. This test can be a pragmatic choice in case a test collections is not created based on pooling, or a test collections is updated or expanded for maintenance purposes, or any other scenario in which the traditional LOU test is not applicable. The fact it relies on simulated judgments makes it attractive as an analytic instrument, as it can provide useful guidance on many of the crucial parameters and decision about pooling prior to relevance assessments—think of the inclusion or exclusion of some runs, or the depth of pooling per eligible system.

Our general conclusion is that the proposed approach to update or expand a test collection offers novel and cost effective ways to build new test collections, and to refresh and update existing test collections. This offers new ways of effective maintenance of test collections for offline evaluation in dynamic domains such as the web. Moreover, the proposed leave-uniques-in test is an effective way of evaluating reusability of different test collections in different test collection building phases, and relies on simulated rather than human relevance judgments, making it particularly attractive for what-if type of analysis prior to committing to the high costs and effort of the relevance assessment stage.

There are some open questions to address in future work. How general can the approach be applied? The case of the TREC contextual suggestion track had a unique configuration with both a fixed offline test collection and judged results from the open web, which greatly facilitated the experiments of this chapter. The general case underlying the approach is dynamic data, such as almost all web data, and the track setup even models this with a crawled web collection from 2012 in combination with live web results from 2014. Our experimental data started with very sparse judgments (ClueWeb12) in combination with a considerable higher number of added pages and judgments (open web), how much is the impact in case the initial test collection was more complete? Web data is highly dynamic, with considerable numbers of new pages appearing in the index continuously making offline test collection age fast [137]. This leads to many high ranked but unjudged pages creating an obvious need to update the offline tests, and ways to reuse old judgments are of obvious value. How sensitive is the approach to the quality of the judgments on the inserted pages? Clearly adding just any labeled data may have some risks, as the judgments may be noisy or made under very different task assumptions, or even give opportunities for spamming [131]. We assume the new and old judgments are created in a similar way, typically by trusted editorial judges or through crowdsourcing platforms as used in this chapter.

One of the impacts of the analysis as reported in this chapter was to add an extra stage to the test collection building efforts in the TREC Contextual Suggestion Track. As the popular open web as data collection presented a considerable factor mitigating

the reusability of the resulting test collection, a multistage stage test collection building approach was adopted at TREC 2015. The TREC 2015 contextual suggestion track [42] started with an early collection building stage where participants contributed any URL to build a fixed "open web" collection that restricted the URLs or pages eligible to be returned. It was followed by a "Live" task in which participants submitted their runs that could only retrieve pages from the earlier collection of URLs. As many participants still restricted their Live runs to particular portals or parts of the collection, a second "Batch" task was added as a reranking task on a given set of URLs extracted from the pools of the "Live" runs. The Batch task of 2015 was added during the track, and proved so popular it became the main task in the TREC 2016 contextual suggestion track [76], in which, as detailed in the previous chapter, we also released the complete crawled content of the whole collection. This "Batch" mode of evaluation circumvented many of the problems analysed in this chapter, and by definition satisfies all pooling effects and reusability tests based on these, as all runs retrieve the exact same set. Although a pragmatic solution to create a reusable test collection, it does so at the cost of experimental power and doesn't solve the fundamental underlying problems analysed in this chapter.

In the last three chapters, we addressed improving user experience in physical spaces such as smart museums and cities by modeling users interacting with smart devices to provide an effective personalized POI recommender system, creating a reusable test collection for personalized POI recommendation offline evaluation, and maintaining reusability of the personalized test collection. In the next two chapters, we focus on modeling users interacting with smart speaker IAs to improve user experience at physical spaces such as smart homes.

5

Impact of Domain and User's Learning Curve on Task and Session Identification in Smart Speaker Intelligent Assistants

In the rest of the thesis, we focus on modeling users behavior on smart speaker intelligent assistants. In this chapter, we address *RQ4: What is the impact of the learning curve and task domain on task and session boundaries when interacting with intelligent assistants?* We study impact of learning phase and domain as contextual factors on users interaction behavior while interacting with intelligent assistants and then estimate task and session boundaries in smart speaker intelligent assistants.

5.1 Introduction

There is a growing interest in integrating Intelligent Assistant (IA) Systems in different devices with an aim of providing enriched experiences for users [21]. For instance, IAs such as Apple Siri, Google Now, Microsoft Cortana and Amazon Alexa have been integrated with Desktop computers, smart phones, and smart speakers. However, user behavior varies in different contexts [76, 100, 155], like platform, input method, etc. For example, users can click on IA responses and change their view-port in interacting with an IA on smart-phones or desktops [109, 169], which is not available in smart speakers. Therefore, due to behavioral dynamics in interacting with IAs, their evaluation on different platforms is challenging, suggesting that different means of evaluation for different platforms may be necessary.

Understanding user behavior and evaluating user satisfaction in interacting with IAs on mobile phones and Desktop computers has previously been studied [94, 103, 109, 110, 119, 168, 169]; however, to our knowledge, there have been no studies investigating user satisfaction and IA effectiveness for smart speakers, which are becoming increasingly popular. For instance, one study found that there was a 128.9% increase in the number of smart speaker users in the United States in 2017 compared to 2016[1]. In this thesis, we use the phrase smart speaker to refer to a wireless speaker device that

[1] https://www.emarketer.com/Article/Alexa-Say-What-Voice-Enabled-Speaker-Usage-Grow-Nearly-130-This-Year/1015812

integrates an intelligent assistant. For the purpose of this study, we focus on devices that have no screen and where the only method of communicating with the device is via voice. Smart speakers can be used for many tasks, such as arranging meetings and controlling home devices via home-automation. This multi-task nature of smart speakers creates a multi-task experience for users, where a task refers to a single goal or information need that the user wishes to satisfy [92]. Furthermore, a series of tasks can be composed to form a session, which refers to a short period of contiguous time spent to fulfill one or multiple tasks [97]. Evaluating the satisfaction of users for tasks and sessions is a critical component of IA evaluation; however, it is not obvious how one should define task and session boundaries for IAs.

Identifying sessions based on user inactivity thresholds as a session timeout is the most common session identification approach in Information Retrieval (IR) [34, 51, 119, 145]. The basic idea is to define an inactivity window that can be used to separate sessions. The idea was first proposed by Catledge and Pitkow [29], in which they use client-side tracking to examine browsing behavior. They reported the mean time between logged events is 9.3 minutes and, by choosing to add 1.5 standard deviation to the mean, they proposed a 25.5 minutes inactivity threshold. Over time, this threshold has smoothed out to 30 minutes. Recently, Halfaker et al. [66] proposed a session identification approach by fitting a mixture of Gaussians and reported 1 hour as an inter-activity time threshold as session boundary being appropriate for most user initiated actions. User inter-activity time is the time difference between two consequent user actions in interacting with an information system. An extension of this work for IAs was presented in [118], where it was shown that the session boundary for an IA on a Desktop Computer was 2 minutes. The experiment was also repeated for Web search and shown to be 24.1 minutes. The differences between these three studies suggests that there is no single session boundary that is applicable across platforms.

Furthermore, previous research has considered the session boundary as a fixed threshold for all IA users. However, in this study, we show that there is no single approach to modeling task and session boundaries. Instead, task and session boundaries are affected by contextual factors such as a user's experience in using the system and the task they are trying to accomplish. Specifically, the multi-task nature of IAs leads to different types of user experiences compared to traditional IR systems. Furthermore, there is often a learning curve associated with being new to an IA. In addition to this, tasks related to some IA domains require a longer time to be fulfilled compared to other domains. Therefore, using a single task and session boundary cut-off over all domains and users expertise levels is not ideal for evaluation.

In this chapter, we study the impact that learning curves and usage domains have on task and session boundary cutoffs. Specifically, we jointly identify task and session boundary by fitting a 3-component Gaussian Mixture Model (GMM) on users inter-activity times in interacting with smart speakers. We focus on smart speakers as they have not been studied before and, as previously mentioned, it is expected that user behavior will differ from that of other platforms. However, our findings are applicable to other platforms as well.

In particular, our main aim in this chapter is to study the research question **RQ4**: *What is the impact of the learning curve and task domain on task and session boundaries when interacting with intelligent assistants?* Specifically, we answer the following

research questions:

1. *How does one effectively measure task and session boundary cut offs in intelligent assistant systems?*

2. *Do user learning curves have an impact on session boundary cut-offs?*

3. *What is the impact of the domain on task and session boundary cut-offs?*

Our contributions include: (1) applying an unsupervised approach using a Gaussian Mixture Model (GMM) with 3 components to jointly identify task and session boundary cut-offs; (2) a detailed study of the impact of the learning curve on task and session boundary cut-offs; (3) an analysis of the impact of usage domain on inactivity thresholds for task and session identifications.

In making these contributions, the rest of the chapter is organized as follows. In Section 5.2, we review related work on task and session boundary identification. The session boundary cut-off estimation based on a GMM is described in Section 5.3. Then, we thoroughly analyze the impact of the learning curve and domain on task and session boundary cut-off in Section 5.4 and 5.5. Finally, we present conclusions and future work in Section 5.6.

5.2 Related Work

User session have been extensively used in IR to develop metrics for web analytics and user behavioral understanding. To create sessions, **three** main group of approaches have been used in the literature, namely, navigation-oriented, query-refinement oriented and time-oriented approaches.

Navigation-oriented approaches take advantage of browsing patterns based on HTTP referrers and URLs associated with each request. Cooley et al. [34] proposed an approach to identify sessions, which is based on detecting the start and end of a session based on navigation behavior of users. The beginning of a navigation behavior (without a referrer) shows the start of a session and the end of a session is a point that the navigational trail can not be traced to a previous request.

Although navigation-oriented approaches are effective in identifying task (addressing a single information need) [129], the complexity of this approach and its developmental focus on tasks over sessions makes them inadequate for session identification [66].

Session identification based on query refinements has also been shown to be only effective in identifying single information need sessions (i.e., task in our definition) [87, 92, 132, 133, 138]. Specifically, Jansen et al. [92] defined a session as “a series of interactions by the user toward addressing a single information need”, which is very similar to the definition of task in our study, which we discuss in more detail in Section 5.3. Jansen et al. showed that the query content is a better signal in identifying tasks compared to a session boundary based on a time-oriented approach.

He et al. [87] and Ozmutlu et al [132, 133] proposed a task identification approach based on detecting topic shifts using lexical query reformulations. Moreover, Radlinski and Joachims [138] proposed an approach to identify the topic relevance of a sequence of queries, which is effective for task identification. However, their extensive focus on tracing user queries in order to determine if they address a single information need limits the use in identifying sessions.

Time-oriented session identification approaches are based on estimating an inactivity threshold between logged user interactions. If there is a long period of inactivity between a user's activities, it is likely the user is no longer active, which leads to ending the session and creating a new session when the user returns. The time-oriented session identification was first proposed by Catledge and Pitkow [29], in which they use client-side tracking to examine browsing behavior. They reported 25.5 minutes inactivity threshold as the session boundary, which has been smoothed out to 30 minutes over time and is the value commonly used in the literature [51, 145].

Although the time-oriented approach has been widely used for session identification, some studies have criticized the effectiveness of the time-oriented approach in identifying sessions [97, 120, 126]. Jones and Klinker [97] proposed a supervised approach for automated segmentation of users' query streams into hierarchical units of search goals and missions and reported that the 25.5 minutes threshold is not effective and performs "no better than random" in identifying search tasks. However, they also reported that the time-oriented approach is more effective for session identification compared to task identification.

On the other hand, Halfaker et al. [66] proposed a session identification approach based on a GMM modeled to fit the within-session and between-session user inter-activity times. In contrast to Jones and Klinker [97], Halfaker et al. [66] showed that the global inactivity threshold is an effective session identification approach and reported 1 hour as an inter-activity time threshold, which is appropriate for most user initiated actions. The main disagreement between these two studies is on task identification, for which Jones and Klinker [97] criticize time-oriented approaches as being ineffective, but not session identification. We adopt an approach similar to Halfaker et al. [66] to jointly estimate task and session boundaries using a mixture of Gaussians fit on users inter-activity times.

Recently, Mehrotra et al. [118] applied a 2-component GMM to estimate session boundary in IAs. The authors showed that the session boundary in Microsoft Cortana on Desktop is much shorter than the common 30 minutes session boundary cut-off in traditional search engines. Our work is similar to the cited work in that we also fit a GMM; however, we show that there is no single appropriate fixed session boundary cut-off for IAs and that the session boundary is dependent on contextual factors, such as user expertise.

The research presented in this study is different from the other time-oriented session identification studies as it empirically shows that the task and session boundary cut-off is not static and fixed for all users. Specifically, in Section 5.4 and 5.5, we show how the user learning curve and task domains impact task and session boundary cut-offs in IAs.

5.3 Session Boundary Cutoff Estimation

This section presents an unsupervised approach for task and session identification using GMMs in order to answer our first research question: *How does one effectively measure task and session boundary cut offs in intelligent assistant systems?*

5.3.1 Definitions

In IR, there are three common ways of defining sessions. A session may refer to: "(1) a set of queries to satisfy a single information need; (2) a series of successive queries; or (3) a short period of contiguous time spent querying and examining results." [66, 97] However, in search engine log analysis literature, it is common to use definition (1) as a task definition, in which a user performs a series of interactions to address a single information need [51, 92].

In IAs, users usually take a sequence of steps with an aim of achieving a goal to solve one or more tasks [109]. Since IAs have the ability to keep context from previous queries, this allows for task chaining where the context of one task can be used as input to the next. Considering the multi-task nature of the IA usage, we therefore define tasks and sessions as follows:

- **Task** is a single information need that can be satisfied by at least one query and one IA generated response.
- **Session** is a short period of contiguous time spent to fulfill one or multiple tasks.

Note, that our session definition is similar to the definition of sessions in [97]. Table 5.1 shows an example of a sequence of user's interactions to fulfill three tasks over the course of two sessions. In this example, using common time-based session boundaries in the IR literature [29, 66, 118] lead to 2 sessions. In the first session, the user is trying to complete two tasks: one for setting an appointment and one for sending a text message to someone. After completing these two tasks, the user had about 1 hour of inactivity before querying the IA for fulfilling the third task on controlling media, which leads to creating the second session.

5.3.2 Fitting Mixture of Gaussians

As previously mentioned, using a time-based threshold has been the most common approach in identifying search sessions [51, 145]. Halfaker et al. [66] proposed a methodology based on GMMs to identify clusters of user activities and argue that the regularity with which these activity clusters appear provide a good estimate of inactivity thresholds for defining sessions. More recently, Mehrotra et al. [118] showed that the 2-component GMM is an effective approach to identify sessions in interactions with IA. They also show that using a 2-component GMM leads to finding a 24.1 minutes session boundary for a well-known commercial search engine, which is close to findings of previous studies [51, 145]

In this chapter, we follow the same methodology that is based on GMMs [66, 118]. However, we focus on jointly identifying tasks and sessions by estimating task and

Table 5.1: An example of a user's sessions in a smart speaker.

User Utterance and System Response	Timestamp
Session 1 Task 1: Calendar review and setting an appointment	
U1: What does my day look like tomorrow?	7/14/2017 7:28:45 PM
S1: You don't have anything scheduled for tomorrow.	
U2: Set an appointment with Mary at 14:30 for tomorrow.	7/14/2017 7:28:53 PM
S2: Alright, I can add appointment with Mary to your calendar for tomorrow at 14:30. Sound good?	
U3: Yes, thanks!	7/14/2017 7:28:58 PM
Task 2: Sending a text	
U4: Text Alex and ask if he is available for a short meeting tomorrow	7/14/2017 7:30:37 PM
S4: Sorry, I can't send messages here. Try the app on your phone or PC.	
U5: Send a message on skype to Alex and ask if he is available for a short meeting tomorrow	7/14/2017 7:31:03 PM
S5: Sending a message to Alex, skype	
Session 2 Task 3: Playing music and controlling media	
U6: Play jazz music on Spotify.	7/14/2017 8:38:05 PM
S6: And now, the playlist Coffee Table Jazz on Spotify.	
U7: Set the volume to 40%.	7/14/2017 8:38:17 PM
S7: I've set the volume to 40%.	

session boundary cut-offs using a 3-component GMM. Jointly identifying task and session boundaries helps having a more accurate Gaussian fits on the inter-activity times of user interactions with the IA, and thus having a more accurate task and session boundary identification. We will detail 2- and 3-component GMM methodologies in the rest of this section.

In order to apply the GMM model to identify the inter-activity type component clusters, we pre-process the users interaction logs of IA usage in order to obtain per-user inter-activity times, which is essential to apply the GMM for identifying tasks and sessions. We plot a histogram based on logarithmically scaled inter-query times in seconds and look for evidence of one or two valleys. We follow Halfaker et al. [66] in using the visual inspection method to set the number of component clusters in the GMM. They proposed a visual inspection based on the number of observed valleys in the users inter-activity times as a better approach to define number of clusters compared to other statistical cluster separation measures like Davies-Bouldin Index (DBI) [38]. In Figure 5.2c, an example of observed valleys in users' inter-activity times histogram is shown by black arrows. After identifying the number of clusters, we fit a K-component GMM [17] on the logarithmically scaled inter-query times via Expectation Maximization. We fit both 2- and 3- component GMMs depending on what we observe in the histogram of inter-query times. In the next section, we describe the use of a 2-component GMM and then follow that with a discussion of when it is appropriate to use a 3-component GMM.

Fitting Mixture of Two Gaussians

The main assumption behind fitting a 2-component GMM for identifying sessions is that the inter-activity times of user interactions contains two component clusters: (1) within session inter-activity times (e.g., the time difference between user query U1 and U2 in Table 5.1); and (2) between session inter-activity times (e.g., the time difference between user query U5 and U6 in Table 5.1). If two clusters have been visually inspected, we fit the 2-component GMM on the logarithmically scaled inter-query times using following Expectation Maximization:

$$f(x, \theta) = \sum_{k=1}^{K} p_k N(x; m_k \sigma_k),$$

in which, $K = 2$ for a 2-component GMM and $N(x; m_k \sigma_k)$ is a Gaussian distribution with mean m_k and standard deviation σ_k. We follow [17, 66, 118] in parameter estimation using Expectation Maximization, where the goal is to maximize the likelihood function with respect to the mixing coefficients, the means and the covariances of the components as the parameters.

E Step: compute the expected values of the posterior probabilities for given parameter values as follows:

$$p^i(k|n) = \frac{p_k^i N(x; m_k \sigma_k)}{\sum_{k=1}^{K} p_k^i N(x; m_k \sigma_k)}$$

M Step: re-estimate the parameters based on the current posterior probabilities:

$$m_k^{i+1} = \frac{\sum_{n=1}^{N} p^i(k|n) x_n}{\sum_{n=1}^{N} p^i(k|n)}$$

$$\sigma_k^{i+1} = \sqrt{\frac{1}{D} \frac{\sum_{n=1}^{N} p^i(k|n) ||x_n - m_k^{i+1}||^2}{\sum_{n=1}^{N} p^i(k|n)}}$$

As previously mentioned, the assumption in fitting a 2-component GMM is that the components represent two main aspects: **within-session** and **between-session** interactivity times. Therefore, if there is an effective fit of the bimodal components, a good estimate of inter-activity time threshold for identifying sessions is the point where the inter-activity time is equally likely to be within the first Gaussian fit (within-session) and the second Gaussian fit (between-session). The reflection of the point on the x-axis, which is the estimate of task or session boundary, is shown by dotted lines in Figure 5.1a and other figures of K-component GMMs.

Fitting Mixture of Three Gaussians

In fitting a 2-component GMM, we assume that user interactivity occurs either **within-session** or **between sessions**. However, the multi-task nature of IA lends itself to three interactivity time behaviors. For instance, a user of an IA may perform a sequence of

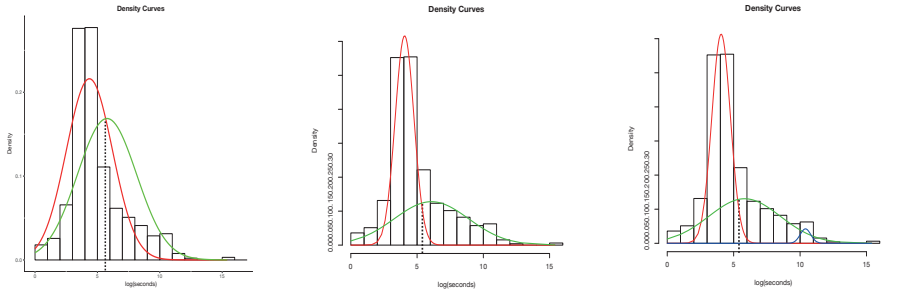
interactions to complete a task followed by a brief pause, which we refer to as **between-task** inter-activity time. They may then complete another task followed by a long period of inactivity. Therefore, we have three inter-activity periods: (1) **within-task** inter-activity times (e.g., the time difference between user query U1 and U2 in Table 5.1); (2) **between-task** inter-activity times (e.g., the time difference between user query U3 and U4 in Table 5.1); and (3) **between-session** inter-activity times (e.g., the time difference between user query U5 and U6 in Table 5.1), noting that the combination of (1) and (2) represent within-session inter-activity time since a session is made up of multiple tasks. Table 5.1 shows an example of a user's sessions having all the above three inter-activity time behaviors. The multi-task behavior of users in interacting with IA motivates us to fit a 3-component GMM on users inter-activity time with an aim of both task and session identification by modeling all the above users inter-activity time behaviors. Jointly modeling both task and session boundary leads to a better fit on users' inter-activity times compared to 2-component GMM, and thus better estimation of task and session boundaries.

Given an effective fit of the three component GMM, we can deduce the following: (1) an estimate of the inter-activity time threshold for identifying tasks is the point where the inter-activity time is equally likely to be within the first Gaussian fit (**within-task**) and the second Gaussian fit (**between-tasks**); (2) an estimate of inter-activity time threshold for session identification is the point where the inter-activity time is equally likely to be within the second Gaussian fit (**between-tasks**) and the third Gaussian fit (**between-sessions**). The reflection of these points on the x-axis is shown by dotted lines in 3-component GMM diagrams, which are estimations of task and session boundaries.

In comparing our approach, the 3-component GMM has been applied in the literature for boundary identification but with a different perspective and application. Halfaker et al. [66] applied a 3-component GMM to model a low-frequency cluster, which represents an extended break corresponding to a life-event with a mode of around 2.5 months. They also fit the 3-component GMM on inter-activity times to have a better fit on the Movielens[2] dataset. They report that in addition to the within-session and between-session interactivity times, they observed an additional component cluster at a high-frequency intervals. They argue that the high-frequency intervals is due to a rapid rating behavior that the Movielens interface allows for. They have observed a similar high-frequency intervals in Movielens searches, for which they stated "we are less sure on how to explain the high frequency component of MovieLens searches. It could be that, unlike when performing a web search (AOL) or reading encyclopedic content (Wikimedia), users' movie searches are more likely to benet from more rapid iteration".

In contrast to Halfaker et al. [66] interpretation, we have shown that the 3-component GMM fitted on user inter-activities is not always about modeling an additional low-frequency or high-frequency clusters to better fit within-session and between-session inter-activity time distributions. Instead, we propose that fitting a 3-component GMM enables us to jointly identifying task and session boundaries. In the most recent work on session identification in IAs [118], a 2-component GMM was used to fit inter-activity times and therfore was not able to fit the three inter-activity clusters that appear in IAs.

[2]https://grouplens.org/datasets/movielens/



(a) Actual Gaussian distributions of users within-task and between-task inter-activity time

(b) EM fit of a 2-component GMM

(c) EM fit of a 3-component GMM

Figure 5.1: Task boundary cut off evaluation based on task boundary crowdsourced labels.

To the best of our knowledge, this is the first study to fit a 3-component GMM to model IA user inter-activity times and, as will be shown in the rest of this section, is often more effective than 2-components models.

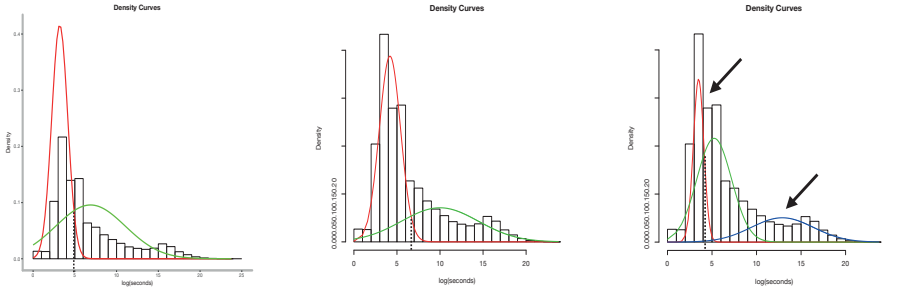
5.3.3 Evaluation

In this section, we first evaluate the effectiveness of GMMs in task boundary identification based on a crowdsourced labeled data. Then, using system-generated task boundary labels, we evaluate effectiveness of the 3-component GMM in identifying within-task distributions. All the experimental results of this section are based on interactions of users with all expertise development level and all available domains.

Evaluation Based on Crowdsourced Labels

To evaluate the effectiveness of the GMM in identifying tasks, we use a dataset of tasks from an IA on desktop computers where the task boundaries were collected through crowdsourcing [119]. The target was identifying the boundary of each task within a session.

To collect the task identification labels for a user session, crowdsource workers judges if the user was trying to find the same information as the previous query by issuing the current query [119]. They could read the user's query or listen to the user's utterances, read or listen to system response, look at the original timestamp of queries, and see a screenshot of a search result page if landing on a search engine result page from the IA. In order to obtain a high-quality task boundary labels, at least 5 crowdsource workers judged each session and the final label is based on a majority vote. The dataset contains 600 IA Desktop sessions, which are divided by judges into around 2000 tasks. Using the crowdsourced labeled data, we can plot the actual Gaussian distribution of within-task and between-task inter-activity times. Therefore, the intersection of the within-task and between-task distributions is the point where the inter-activity time is equally likely to be in either component and is therefore taken as the task boundary.



(a) Actual Gaussian distributions of users within-task and between-task inter-activity time

(b) EM fit of a 2-component GMM

(c) EM fit of a 3-component GMM

Figure 5.2: Task boundary cut off evaluation based on the system task boundary labels.

Figure 5.1a shows the within-task and between-task inter-activity time distributions based on the crowdsourced labels, in which the task boundary is $2^{5.5} \sim 45$ seconds. Furthermore, Figures 5.1b and 5.1c show the user 2- and 3-component GMM fits of inter-activity times based only on the inter-activity times and not any labeled data.

The experimental results show that the intersection point of within-task and between-task Gaussian fits for both the 2- and 3-components GMMs is $2^{5.4} \sim 42$ seconds. The task boundary estimation based on the GMMs is very close to the 45 seconds actual task boundary based on the labeled data. Therefore, in both the case of the 2- and 3-component GMMs, the data suggests that fitting a Gaussian provides a reasonable approach for modeling task boundaries. Note, the data used in this study was sampled at the session level [118]. Therefore we do not have multiple sessions per user, which makes it impossible to model between-session inter-activity times. This explains why the 2- and 3-component GMMs lead to the same task boundary since the GMMs only need to model the within-task and between-task components. The dataset used in the next section contains multiple sessions per user, which allows us to compare the effectiveness of the 2- and 3-component models.

Evaluation Based on System Task Boundary Labels

In the previous section, we showed the effectiveness of GMMs in identifying tasks. In this section, we evaluate the effectiveness of the 3-component GMM in fitting inter-activity times compared to the 2-component GMM. To achieve this, we make use of a high-quality task boundary classifier being used in a commercial IA. The commercial IA provides services for a variety of tasks and it can trace a user's interactions toward fulfilling a task, with the aim of identifying the task completion status, such as *completed*, *in-progress* and *canceled*. Using the task completion status, the IA can identify task boundaries of user interactions, which we call "system task boundary". As was the case with the crowdsourced data, we do not have access to a session completion status in the IA. However, we do have access to multiple sessions per user, which allows us to model the between-session inter-activity times. However, for our evaluation we only focus

measuring within-task and between-task inter-activity times, which is available based on the system task boundary. Due to the fact that we do not have access to the system session boundary as it is defined in this chapter, we do not evaluate the between-session inter-activity times.

To evaluate the 3-component GMM in modeling within-task and between-task inter-activity times, we sampled about 300K queries issued in a two months period of a commercial smart speaker usage. Figure 5.2a shows the IA identified within-task and between-task inter-activity time Gaussian distributions based on the system task boundary labels. The intersection of the within-task and between-tasks Gaussians based on the system task boundary leads to a $2^5 \sim 30$ seconds boundary as the task boundary. Figures 5.2b and 5.2c show the fit of 2- and 3-component GMMs on the inter-activity times without using the system task boundary labels. The intersection point of the within-task and between-task distributions of the 2-components GMM leads to $2^{6.8} \sim 111$ seconds threshold as the task boundary. The intersection point of within-task and between-task distributions of the 3-components GMM estimates $2^{4.3} \sim 20$ seconds as task boundary cut-off. According to this result, the difference between 3-components GMM task boundary estimation and the system task boundary is 10 seconds, which shows the 3-component GMM based task boundary estimation is a more accurate approach to estimate task boundary cut-offs compared to the 2-component GMM based task boundary estimation with a 81 seconds difference.

We also measure the KL-divergence of the system task boundary labeled data distribution and the GMM fit via expectation maximization. KL-divergence is a similarity measure of two distributions and KL-divergence of two Gaussian distributions is given by [123]:

$$
\begin{aligned}
KL(p,q) &= -\int p(x)\ log\ q(x)\ dx + \int p(x)\ log\ p(x)\ dx \\
&= \frac{1}{2} log\ (2\pi\sigma_2^2) + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}(1 + log 2\pi\sigma_1^2) \\
&= log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2},
\end{aligned}
$$

where, σ_1 and σ_2 are the standard deviations of the first and second Gaussian distributions, and μ_1 and μ_2 are the means of the first and seconds Gaussian distributions. A smaller KL-divergence value indicates a more similar Gaussian distributions. The KL-divergence of the system task boundary labeled within-task Gaussian from the within-task Gaussian fit of the 2- and 3-component mixture models are 0.4917 and 0.2544, respectively. According to this result, compared to the 2-component GMM, the 3-component GMM leads to a more effective and accurate within-task Gaussian distribution of user inter-activity times.

To summarize this section, we presented results of fitting 2- and 3-component GMM for task and session identification. The evaluation results show that the 3-component GMM leads to more accurate Gaussian fits of users inter-activity times and a more precise task boundary cut-offs compared to the 2-components GMM. Thus, in the rest of this chapter, we use 3-component GMM to study the impact of domain and user learning-curve on the task and session boundaries. In the next section, we investigate

the impact of learning curve on task and session boundaries by segmenting users by their levels of expertise.

5.4 Impact of learning-curve on Session Boundary Cut-off

This section studies the impact of the learning-curve on session boundary cut-offs, aiming to answer our second research question: *Do user learning curves have an impact on session boundary cut-offs?* We begin by describing our data and then define the learning-curve in user behavior when interacting with an IA. We then discuss how session boundaries differ for the learning-curve and a so called normal usage phase.

5.4.1 Experimental Data

This study is based on two random samples of users interaction logs with two different commercial IAs being used on smart speakers. In the rest of the chapter, we refer to them as Dataset 1 and Dataset 2. Both Dataset 1 and Dataset 2 are based on user interactions with the speaker from the first day they start using it. This enables us to evaluate the impact of learning-curve on session boundary cut-offs.

Dataset 1 consists of interaction logs of 2,087 users collected from March 2017 to September 2017. Users of Dataset 1 issued 731,128 queries in this period, which is 350 queries per user on average. Dataset 1 contains a total of 644,225 tasks. The dataset has query timestamps and domain classifications of the queries.

Dataset 2 consists of interaction logs of 20 users with an average usage period of 264 days. The dataset includes 69,649 queries, which is 3,482 queries per user on average. Although the Dataset 2 has fewer users, the average number of queries per user is larger than Dataset 1. Furthermore, the queries span a larger time frame, which enables a longer term analysis.

5.4.2 Learning-curve Definition

According to research in library search and search engines [50, 51, 84, 113, 165], domain expertise enhances search performance, and the development of search expertise over time has been observed in prior studies [157, 165, 166]. Specifically, White et al [165] showed that non-expert domain expertise develop over time.

In this chapter, we focus on the impact of the learning-curve phase on IA task and session identification. Smart speakers often provide a new experiences for users who are not familiar with them. Therefore, users who are new to using smart speakers are generally non-experts and curious to interact with the IA, which motivates them to query the smart speaker more frequently when they first start using it compared to the normal usage. We name this stage of the new user usage as the learning-curve phase, in which users try to learn the device functionality and satisfy their curiosity. In this chapter, the period after the learning-curve phase is named normal-phase. The difference in users' behavior in learning-curve and normal phase has been also observed in Figure 5.3. Figure 5.3 shows impact of learning-curve on average number of queries issued per

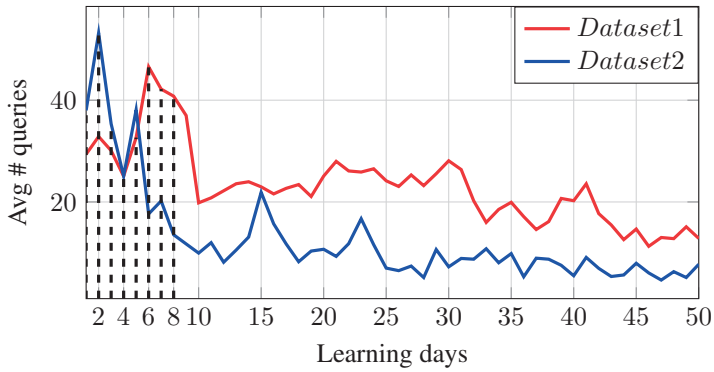
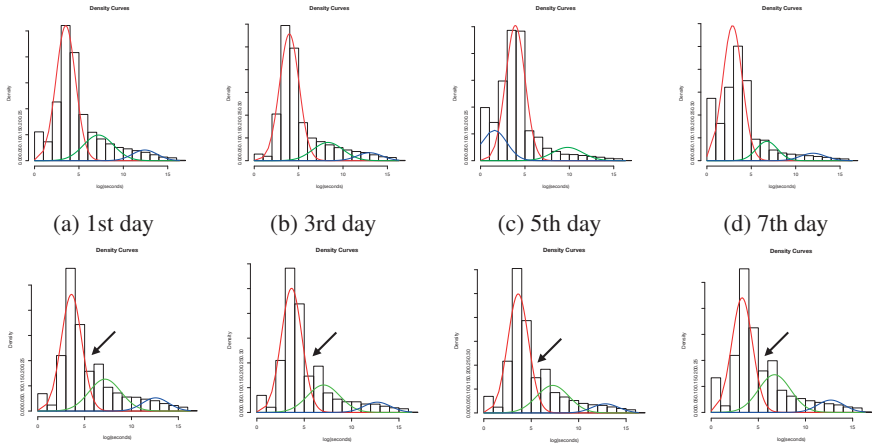


Figure 5.3: Impact of learning on average number of queries per day.



(a) 1st day (b) 3rd day (c) 5th day (d) 7th day

(e) 9th day (f) 11th day (g) 13th day (h) 15th day

Figure 5.4: Dataset 1 users' interactivity time distribution from 1st day to 15th days of their usage.

day. Learning-curve is modeled by users' expertise based on number of usage days in this diagram.

Figure 5.3 indicates that the average number of queries issued by users drops after 8 usage days with no significant increase in average number of queries after the 8th day in both Dataset 1 and Dataset 2 interaction logs. For example, average number of queries drops from 42.18 and 40.79 in seventh and eighth days, respectively, to 36.99 and 19.83 average number of queries in the ninth and tenth days for Dataset 1 users. We observed a similar pattern, yet with less considerable drops, for Dataset 2. Specifically, average number of queries in Dataset 2 is 22.2, 13.55, 11.75, and 9.95, for days 7, 8, 9, and 10, respectively.

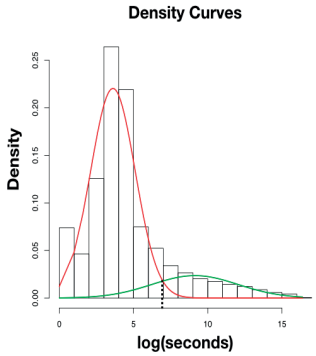
5.4.3 Identifying Session Boundary Cutoff in learning-curve

In addition to our observation in Figure 5.3, Figure 5.4 shows a per day basis analysis of users inter-activity times in Dataset 1. Plotting the inter-activity times of new users in their first fifteen days of using smart speakers leads to a histogram where there is no evidence of a trimodal Gaussian distributions in users' inter-activity times in first eight days. However, in this experiment, we have observed that after the eighth usage day, an additional valley appears in the histogram, which is shown by a black arrow. According to the observations in Figures 5.3 and 5.4, we chose to consider any user interactions logged in the first eight days of their usage as part of the learning-curve phase, in which users are curious and issue many queries. The rest we define as their normal phase. We know that using a 8 days learning-curve phase based on our observations is a strong assumption. For example, we could have chosen 7 or 9 days as the learning-curve phase. Identifying an accurate learning-curve phase is not the main focus of this experiment. In this experiment, we are interested in showing that the task and session boundaries are different in learning-curve phase compared to the normal-phase. In the rest of this section, we detail task and session boundary cut-offs on Dataset 1 and Dataset 2 in users learning-curve compared to their normal phase.

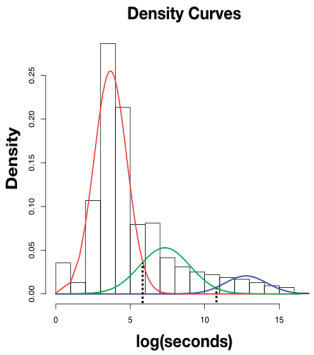
In Figure 5.5, we have plotted histograms of Dataset 1 and Dataset 2 users inter-activity times in their learning-curve as well as their normal phase. In Figures 5.5a and 5.5c, we fit a 2-component GMM on user inter-activity times in their learning-curve to estimate task boundary cut-off. The intersection point of the within-task and between-task Gaussian distributions (task boundary) of the learning-curves are $2^{6.9} \sim 119$ and $2^{6.3} \sim 79$ seconds for Dataset 1 and Dataset 2 users, respectively, which are similar to the 2 minutes task boundary cut-off of Microsoft Cortana on Desktop [118]. During the learning-curve phase, as we do not observe a tri-modal Gaussian distribution in user inter-activity times shown in Figures 5.5a and 5.5c, there is not any clear evidence of sessions in user inter-activity time distribution. One possible explanation of it is that users are more curious to try the IA to learn its functionality in their learning-curve phase rather than querying the IA in a session-based scenario to fulfill one or more information needs.

Furthermore, Figures 5.5b and 5.5d show a fit of the 3-component GMM on the normal-phase inter-activity times. The intersection point of the within-task and between-task inter-activity times distribution (the task boundary cut-off) is $2^{5.8} \sim 56$ and $2^{6.5} \sim 91$ seconds for Dataset 1 and Dataset 2, respectively. In addition, we identify the session boundary cut-off based on the intersection point of the between-task and between-session inter-activity times distribution, which is $2^{10.8}\ seconds \sim\ 30\ minutes$ and $2^{11.5}\ seconds \sim 48\ minutes$ for Dataset 1 and Dataset 2, respectively.

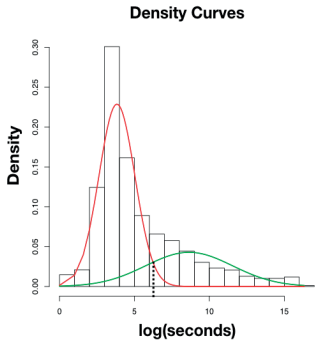
Although the 48 minutes inactivity threshold to identify IA sessions seems long compared to the common 30 minutes session boundary cut-off in search engines, we suspect that the smart speakers domains of usage might contribute in this difference. Specifically, many of the queries are related to domains like controlling media and listening to music in smart speakers, which requires a longer session duration compared to the typical search engine sessions. Figure 5.6 shows the top-5 popular domains being used in Dataset 1. Apart from the control media, the rest of the top-5 popular domains are short-term tasks based on number of queries per task. That is one of the possible



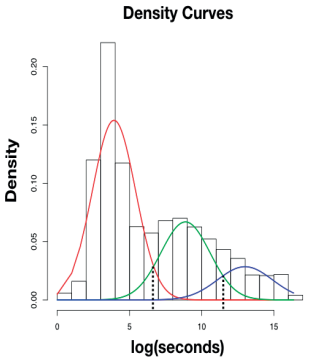
(a) Learning-curve for Dataset 1 users



(b) Normal phase for Dataset 1 users



(c) Learning-curve for Dataset 2 users



(d) Normal Phase for Dataset 2 users

Figure 5.5: Impact of learning-curve on task and session boundary. The difference between mean of the learning-curve and the normal-phase inter-activity times distributions is statistically significant based on t-test ($\rho < 0.05$).

explanations why the task and session cut-offs in Dataset 1 are relatively short.

To summarize, in this section, we studied the impact of a learning-curve on session boundary cut-off and showed that there is not any evidence of sessions having multiple tasks during the learning-curve phase. By contrast, during the normal-phase, we observed evidence of both tasks and sessions in users inter-activity times.

5.5 Impact of Usage Domain on Session Boundary Cutoff

The final question we address in this chapter is how different domains and tasks affect session boundaries. We investigate whether there is a dependency between session boundary and the domain of a user's information needs. This section answers our third research question: *What is the impact of the domain on task and session boundary cut-offs?*

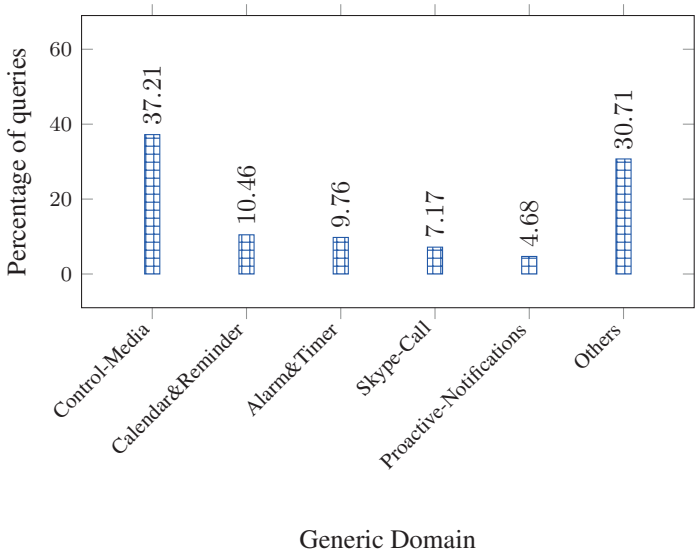
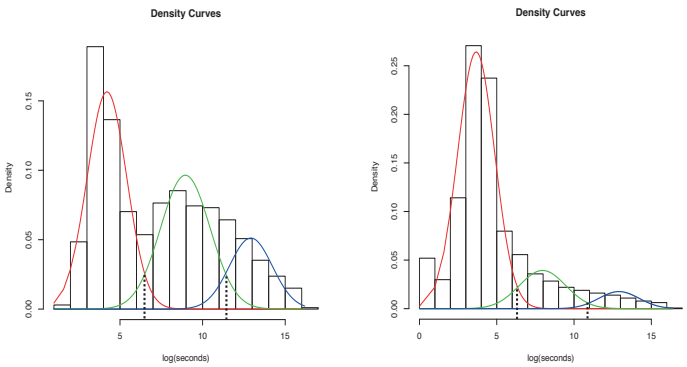


Figure 5.6: The top-5 popular generic domains in dataset1.



(a) Control-media domain (b) All domains

Figure 5.7: Impact of domains on session boundary. The difference between mean of the control-media and all domain inter-activity times distributions is statistically significant based on t-test ($\rho < 0.05$).

As it is shown in Figure 5.6, about 37 % of the smart speaker usage in Dataset 1 is in control-media domain. Therefore, we focus on this domain and study how task and session boundaries in this domain differ from those of other domains. The control-media domain contains queries with intents, such as query media, play music and volume up. In this experiment, we preprocessed the inter-activity times for each day of data. Specifically, we analyzed user queries each day and, if all the queries of a user's interactions in a day is in control-media domain, we consider the inter-activity times of the user in the day as control-media inter-activity times. Otherwise, the user's

inter-activity times in the day is in all domains category. The control-media inter-activity times is 11.6 % of the all user inter-activity times. All the available domain labels of users' queries available in Dataset 1 come from a high-quality commercial domain classifiers.

Figure 5.7 shows a fit of three Gaussians on users' inter-activity times of control-media domain against all domains together. As it is shown in Figure 5.7a, in control-media domain usages, the task boundary cut-off is $2^{6.5} \sim 91$ seconds and the session boundary cut-off is $2^{11.5}\ seconds \sim\ 48 minutes$. In contrast, the 3-component GMM fit on all domains inter-activity times lead to $2^{6.3} \sim 79$ seconds task boundary cut-off and $2^{10.9}$ seconds $\sim$ 32 minutes session boundary cut-off estimations.

These results indicate that estimating task and session boundary cut-offs for all domains may be misleading. Instead the data suggest that task and session boundary cut-off identification should be done per domain or on a group of similar domains. Furthermore, we have observed that the difference between the mean of control-media inter-activity times distribution and the mean of all domain inter-activity times distribution is statistically significant based on t-test with $\rho < 0.05$, which supports our conclusion in this section.

5.6 Discussion and Conclusions

In this section, we will first discuss impact of our study in downstream applications and computational costs of our proposed task and session identification model. We then briefly detail conclusions.

5.6.1 Discussion

Impact on Downstream Applications

Task and session identification is a key element of many IR applications such as evaluation studies based on user interaction logs, user modeling, and personalization. Specifically, in user satisfaction prediction based on implicit signals from user interactions, which is an emerging metric to evaluate Web search engines [84], task and session identification is potentially having a direct impact on effectiveness of a user satisfaction classifier.

Table 5.1 shows an example of a user session in a smart speaker. In task 2 of this example, if a task identifier indicates that the task is terminated after system response to the user's fourth query (U4), an effective user satisfaction classifier would most likely classify the task as a dis-satisfactory (DSAT) task. However, if a task identifier indicates system response to the user's fifth query (U5) as the end of the task, the effective user satisfaction classifier would most likely classify the task as a satisfactory (SAT) task. Session identification would also have a similar impact on session-level user satisfaction prediction, for which we do not provide an example because of space limitation in this chapter. Our main point is that the task and session identification could have a direct impact on user interaction log based studies such as user satisfaction prediction problem.

Cost

our proposed task and session identification model is a time-oriented approach and time-oriented approaches calculate task or session boundary once off-line then use it on-line without any re-estimation, it is computationally cost-effective compared to navigation-oriented and query-refinement oriented approaches. In fact, we just need to estimate task and session boundary once and then use it for a downstream application, which is almost as cheap as using the 30 minutes session boundary for session identification in search engine query logs.

Contextual Factor

In this chapter, we have studied impact of contextual factors on task and session boundaries. We decided to focus on domain and learning-curve because our observations shows that they are important contextual factors to be aware of while modeling users' behavior. They were also easy to measure. Our main aim was to show that contextual factors do matter and these are just two examples of them. We agree that additional study is required to investigate other contextual factors such as location, which we leave for future works.

5.6.2 Conclusion

In this chapter, we investigated the impact of the learning-curve and usage domain on task and session boundary cut-off for two different IAs being used in smart speakers. We experimented with an application of a 3-component Gaussian mixture model to fit user inter-activity times with the aim of jointly identifying both task and session boundary cut-offs in IA user interaction logs. Our main research question was: *What is the impact of the learning curve and task domain on task and session boundaries when interacting with intelligent assistants?* Specifically, we answer following research questions:

Our first research question was: *How does one effectively measure task and session boundary cut offs in intelligent assistant systems?* We evaluated 2- and 3-component GMMs in task and session boundary estimation based on crowdsourced task boundary cut-off labels and system generated task labels. Our results show that fitting a GMM on user inter-activity times is an effective approach to estimate task and session boundary cut-offs in IA usage on smart speakers. Furthermore, our experimental results show that using a 3-component GMM leads to a better estimation of task boundary cut-offs compared to a 2-component GMM.

Our second research question was: *Do user learning curves have an impact on session boundary cut-offs?* We showed how an additional inter-activity time cluster appears in normal phase, which is not available in learning-curve phase. We concluded that while using 2-component GMM leads to a reasonable fit of users inter-activity times in their learning-curve phase, fitting a 3-component GMM is more effective during the normal-phase. In fact, there is not significant evidence of sessions having multiple tasks in learning-curve phase, and users tend to accomplish tasks more frequently in learning-curve phase compared to normal-phase of their usage.

Our third research question was: *What is the impact of the domain on task and session boundary cut-offs?* According to the experimental results, task and session boundaries differ across domains and therefore the domain of a task should be considered when measuring these boundaries.

In summary, our general conclusion of this chapter is that task and session boundary cut-offs are not static but are instead dependent on contextual factors like the user's learning curve and their usage domains. In the next chapter, we use the estimated task boundary to identify tasks from the smart speaker interaction logs, and then predict user satisfaction on the identified tasks.

6

Measuring User Satisfaction on Smart Speaker Intelligent Assistants Using Intent Sensitive Query Embeddings

To improve the user experience in interacting with smart speakers at a physical space such as smart homes, in this chapter, we address *RQ5: How to evaluate user satisfaction in Intelligent Assistants based on user queries?* We focus on using sequence of user's utterances to model users behaviors interacting with smart speaker intelligent assistants to measure their satisfaction in performing a task.

6.1 Introduction

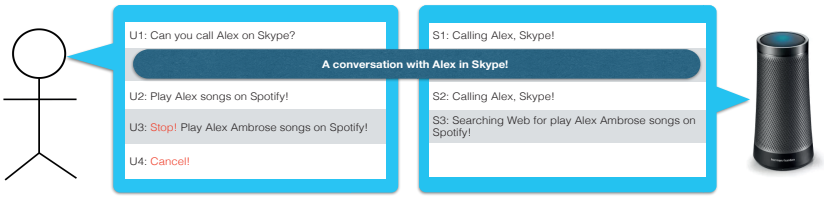
There is a growing interests in integrating intelligent assistants (IAs) such as Apple Siri, Google Now, Microsoft Cortana and Amazon Alexa in different devices [21]. This has led to the creation of smart devices, such as smart phones and smart speakers. Each of these smart devices provides device specific means of interaction. For example, smart speakers do not include any screen interface and therefore users interact with them via voice. This is in contrast to IAs on Desktop computers and mobile phones, which have clicks and gestures as user interactions [119, 169]. Since user behavior differs on these different platforms and in different contexts [68, 76, 100, 155] there may be a need to develop different means of evaluation for different platforms.

Smart speaker devices with integrated IAs such as Amazon Echo, Google Home, Apple Homepod, and Harmon Kardon Invoke with Cortana have become increasingly popular in recent years. As we mentioned before, one study found that there was a 128.9% increase in the number of smart speaker users in the United States in 2017 compared to 2016[1]. Therefore, measuring the effectiveness of IAs on these popular smart devices is becoming increasingly important.

An emerging metric for evaluating Information Retrieval (IR) and IA systems is user satisfaction, which is often based on user interaction data [2, 54, 83, 85, 95, 105, 107, 109, 110]. User satisfaction is a subjective measure of a user's experience with an

[1] https://www.emarketer.com/Article/Alexa-Say-What-Voice-Enabled-Speaker-Usage-Grow-Nearly-130-This-Year/1015812

(a) SAT session example.



(b) DSAT session example.

Figure 6.1: SAT/DSAT user session example in smart speaker IA.

information system, which indicates to some extent if the user's desire or goal is fulfilled [102]. User satisfaction evaluation in IAs on mobile phones and Desktop computers has previously been studied [94, 103, 109, 110, 119, 168, 169]; however, to our knowledge, there have been no studies investigating user satisfaction and IA effectiveness for smart speakers. In this chapter, we use the phrase smart speaker to refer to a wireless speaker device that integrates an IA. For the purpose of this study, we focus on devices that have no screen and where the only method of communicating with the device is via voice.

Many implicit signals have been studied for measuring user satisfaction in Web search or for IAs on desktops and mobile devices. Some examples of signals include: clicks followed by a long dwell-time [55, 84, 95, 106, 107], mouse movements [119], touch gestures [109, 169], and browser view-port interactivity [111]. However, besides user queries, none of these implicit user satisfaction signals are available for smart speakers due to voice being the only method of interaction. Therefore, evaluating user satisfaction with IAs on smart speakers presents a new challenge, which is finding an effective implicit user satisfaction feedback signal.

Since queries are the only means by which users interact with smart speakers, they represent a natural starting point for measuring user satisfaction. In fact, the reformulation of a query is a well known signal for user dissatisfaction [86]. In this sense, one can think of queries as not only representing a user's information need, but also as providing an implicit feedback signal similar to the case of a click in Web search or a touch on mobile phones. In this chapter, a query is considered more generic than the typical query used in IR literature. Specifically, in this study, queries could be typical Web queries or command-like queries (e.g., device control commands). Figure 6.1a shows an example of a SAT session, in which a user tries to complete a "setting an appointment" task after making a call to Alex. We hypothesize that the sequence of user queries can be a beneficial implicit user satisfaction (SAT) signal. Specifically, issuing the query "Yes, thanks" after the query "Set an appointment with Alex at 14:30 for tomorrow." might be an indicator of user satisfaction. However, the user saying "Cancel" after the query might be an indicator of user dissatisfaction (DSAT).

Furthermore, Figure 6.1b indicates an example of a DSAT session. In this session, a user asks the smart speaker to play a song by Alex on Spotify after making a call to Alex (i.e., U2). However, the IA does not detect the user intent properly as it wrongly determines that the context is still making a call, which leads to an undesirable system response. As the system response is not satisfactory, the user tries to stop the IA from giving a wrong response and issues a similar query again (i.e., U3). The system responds by searching the Web for the user query, which is not satisfactory for the user as she has stopped the smart speaker by saying "Cancel". As can be seen by these examples, a sequence of queries can lead to an implicit feedback signal of user satisfaction. The question then arises on how one should use these queries to measure user satisfaction and, as the specific focus of this chapter, how one can find effective query representations for measuring user satisfaction.

In this chapter, we hypothesize that the intent of a user query can function as an implicit signal for measuring user satisfaction, where intent refers to the meaning of the query [139]. For instance, the query "Set an appointment with Alex at 14:30 for tomorrow." has "create_calendar_entry" intent. If this is followed by a "confirm" intent, then we may conclude the user was satisfied. In contrast, if it was followed by a "reject" intent then we may conclude the user was dissatisfied. Based on this intuition, we propose to measure user satisfaction based on representations of user queries that are intent sensitive. We propose to do this in two ways. In the first way, we define Intent Sensitive Word Embeddings (ISWEs), which are word embeddings that not only represent the semantics of words, but also semantics of the intents associated with words. For example, although the queries "Yes, thanks" and "Cancel" occur in similar word contexts, (e.g., "Set an appointment with Alex at 14:30" → "Yes, thanks!" and "Set an appointment with Alex at 14:30" → "Cancel"), they have very different intents, i.e., "confirm" for the former and "reject" for the latter. Our proposed methods for producing ISWEs scatters these words with different intents in the representation space. We use this sequence of ISWEs to measure user satisfaction based on a series of user queries.

In the second approach, we consider each query as a single unit having a single intent and train query representations based on a sequence of query intents. For example, "play some jazz music" is a single query having a single intent of "play_music", and "Set an appointment with Alex at 14:30" → "Yes" is an example of a task containing two queries with intents "create_calendar_entry" and "confirm". Therefore, these would represent sequences of length 1 and 2, respectively, and we use these sequences of intents as input to our proposed user satisfaction prediction model. This approach differs from our other proposed approach since in ISWE we use intents to derive intent sensitive word embeddings. In this approach we forgo the words and only focus on the intent of the entire query.

In this chapter, our main aim is to study the research question **RQ5**: *How to evaluate user satisfaction in Intelligent Assistants based on user queries?* Specifically, we answer the following research questions:

1. *How to model intent-sensitive query representations for user satisfaction prediction?*

2. *How effective is the proposed intent-sensitive user satisfaction model in evaluat-*

ing intelligent assistants on smart speakers?

Our contributions include: (1) a user satisfaction prediction model that predicts user satisfaction based on just user query sequences; (2) proposing a novel intent-sensitive word embedding (ISWE) that can capture query term intents by learning word representations based on both word neighbor context and query intent; and (3) an unsupervised intent embedding approach based on the Skip-gram model that learns intent representations for each query.

In making these contributions, the rest of this chapter is organized as follows. In Section 6.2, we review related work. Section 6.3 is devoted to task and user satisfaction definitions. The proposed user satisfaction prediction model and intent-sensitive query representation learning methods are described in Section 6.4. Section 6.5 presents experimental results of our study. Finally, we present our conclusions and discuss future work in Section 6.6.

6.2 Related Work

Related work falls into two categories: we first review user satisfaction in search and intelligent assistants, and then we discuss related work on word embeddings.

6.2.1 User Satisfaction

Online evaluation have been widely used to control and improve IR system effectiveness [13, 46, 48, 154]. An emerging metric for evaluating Web search engines is user satisfaction based on implicit signals from user interactions [2, 54, 83, 85, 95, 105, 107, 109, 110]. User satisfaction in search is a subjective measure of a user's search experience, which is addressed by the extent to which a user's specified desire or goal is fulfilled [102]. User satisfaction is different from traditional relevance measures in IR such as MAP and Precision, which are based on relevance of the retrieved results for a given query. In user satisfaction, user experience and their success in fulfilling a goal plays a major role, which has been addressed based on user interaction signals in web search [83, 86, 105, 107] and intelligent assistants [109, 110, 119, 168, 169].

One of the common signals that has been used for user satisfaction prediction is a click followed by a long dwell-time [55, 84, 95, 106, 107]. Hassan et al. [86] propose query reformulation as a signal of user dissatisfaction and they show that incorporating query features and query reformulation in user satisfaction prediction outperforms an approach based on click features alone.

Session and SERP features such as time to the first click and average number of clicks per query have been also studied in personalized and customized search satisfaction prediction [84]. Furthermore, gesture features, such as reading time and touch actions, have been used in search satisfaction prediction [109] and good abandonment detection [169] in mobile web search. In addition, tracking the browser viewport on mobile devices has been also studied as an implicit signal for user search satisfaction [111].

To model user satisfaction, Hassan et al. [85] model the search process as a sequence of actions, such as queries and clicks, and built two Markov models for identifying satisfactory and dissatisfactory search sequences. They further shows that using a semi-supervised search success prediction approach based on sequence of actions can lead to an improvement over the supervised approach [83]. More recently, Mehrotra et al. [119] proposed user satisfaction prediction in Desktop intelligent assistants based on fine-grained actions, such as mouse movements.

Our work is different from all the above user satisfaction prediction models in two aspects. First, we focus on user satisfaction prediction with IA on smart speakers, in which the only means of interaction is via voice queries. In fact, none of the past works are applicable in user satisfaction on smart speakers, except the query reformulation proposed in [86]. Our proposed approach is different from the query reformulation as we use a sequence of query terms to predict user satisfaction. Furthermore, we propose user intent as a new signal to measure user satisfaction.

6.2.2 Word Embeddings

Recently, continuous word embeddings have gained popularity in different IR tasks, such as query and document language models [58, 177], neural ranking models [45, 178, 180], and query expansion [7, 47, 176]. In particular, Zamani and Croft [176] propose a theoretical framework for query embedding vectors representation based on individual vocabulary term embeddings. Furthermore, they propose using word embeddings to weight terms that do not occur in the query, yet are semantically related to the query terms in the query language model. More recently, Ai et al. [3] proposed a hierarchical embedding model that jointly learns distributed representations for query, product and users in a personalized product search.

Zamani and Croft [178] recently showed that the linear context is not sufficient for learning an effective word embeddings for IR tasks, and they propose learning word representations based on query-document relevance information. In addition, Rekabsaz et al. [141] propose post filtering of related terms by global context relatedness measures to avoid topic shifting in retrieval models. Furthermore, Mehrotra and Yilmaz [117] propose learning query representations based on task context in search logs.

Although, incorporating additional signals to improve word embeddings in IR is very new, there have been plenty of research in NLP to improve word representations by using metadata [184], semantic lexicons [135], syntactic word relations [114] and document topics [53]. Our work is different from the above as we propose a novel Intent Sensitive Word Embedding (ISWE) method that can leverage information from a query's intent to improve query term representations. We are the first who modify the Skip-gram model [122] to capture query term intents and use them as input to a user satisfaction prediction model.

6.3 Task Satisfaction

In this section, we first define tasks and sessions as they apply to IAs, and then we define task satisfaction in IAs. In IAs, users usually take a sequence of steps to achieve a goal

Table 6.1: An example of a user's task satisfaction in an IA on a smart speaker.

User Utterance and System Response	User Satisfaction
Task 1: Calendar review	
U1: What does my day look like tomorrow? S1: You don't have anything scheduled for tomorrow.	SAT
Task 2: Sending a text	
U2: Text Alex and ask if he is available for a short meeting tomorrow S2: Sorry, I can't send messages here. Try the app on your phone or PC.	DSAT
Task 3: Calling on Skype	
U3: Can you call Alex on Skype? S3: Calling Alex, skype	SAT
Task 4: setting an appointment	
U4: Set an appointment with Alex at 14:30 for tomorrow. S4: Alright, I can add appointment with Alex to your calendar for tomorrow at 14:30. Sound good? U5: Yes, thanks!	SAT

to solve one or more tasks [109]. Since IAs can keep context from previous queries, this allows for task chaining where the context of one task can be used as input to the next. Considering the multi-task nature of the users' behaviors in IA, we follow the task and session definitions as proposed in [80]:

- **A Task** is a single information need that can be satisfied by at least one query and one IA generated response.
- **A Session** is a short period of contiguous time spent to fulfill one or multiple tasks.

Given the definitions of tasks and session, we define task satisfaction as follows:

- **Task satisfaction** is how successful a user is in completing a single information need using at least one query and receiving at least one IA generated response.

IA generated responses are not always in the form of replying to a user query in a dialogue manner. For some queries such as "Stop", a proper IA response could be simply stopping whatever the IA was doing. Table 6.1 shows an example of a user's task satisfaction in an IA session. In this example, the user is performing four tasks, including: reviewing her calendar; sending a text; calling on Skype; and setting an appointment. These tasks are part of a session, in which the user is organizing a meeting with Alex. Tasks 1 and 2 in Table 6.1 show examples of satisfactory (SAT) and a dissatisfactory (DSAT) tasks, respectively.

To summarize, in this section, we have defined task and sessions in IAs, and then we have defined task satisfactions in IAs. In the next section, we detail the task satisfaction prediction problem in smart speaker IAs and describe our proposed task satisfaction prediction model.

6.4 Task Satisfaction Prediction

This section first presents user satisfaction prediction problem based on a sequence of user queries. We then detail our proposed user satisfaction prediction and query representation learning models.

6.4.1 Satisfaction Classification Model

The task of user satisfaction prediction based on a sequence of user queries can be regarded as a sequence classification problem. To be more specific, a user starts querying the IA at time stamp t_0 and can keep querying the IA up to time stamp t_n in a task or a session. Therefore, we can represent a user's set of interactions with an IA on a smart speaker as a sequence of queries $q_{t_0}, q_{t_1}, q_{t_2}, ..., q_{t_{n-1}}, q_{t_n}$, where q_t is a query q at time stamp t. Given a sequence s of queries $q_t \in Q$, the task is to predict whether the sequence of queries leads to a satisfactory (SAT) or a dissatisfactory (DSAT) experience in a task. In particular, using a variable $c \in \{0, 1\}$, the goal is to find the most likely class c, given a sequence s.

Considering the sequential nature of user queries and its variable length in accomplishing tasks in IA, we propose to use a Long Short Term Memory (LSTM) Recurrent Neural Network (RNN) to model user satisfaction because of the following reasons: 1) LSTMs have been shown to be effective in different sequence classification problems such as text classification [60], sentence similarity [128] and satisfaction prediction in the case of good abandonment [167]. 2) LSTMs are more effective than standard RNNs in their ability to model long time dependencies.

The LSTM updates a hidden layer representation sequentially using time steps relying on four components: 1) a memory state c_t, 2) an input gate i_t, 3) a forget gate f_t, and 4) an output gate o_t. The input and forget gates control what gets stored in memory based on each input and the current state. The output gate controls how the memory state impacts other units. In an LSTM, updates at each time step t are as follows:

$$
\begin{aligned}
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
\widetilde{c_t} &= tanh(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= i_t \odot \widetilde{c_t} + f_t \odot c_{t-1} \\
h_t &= o_t \odot tanh(c_t),
\end{aligned}
$$

where x_t is an input at time step t, $\widetilde{c_t}$ is the candidate value for the state of the memory cell, and h_t is the output of the unit. The W_i, W_f, W_o, and W_c are the weight matrices for the current input. The U_i, U_f, U_o, and U_c are the weight matrices for the previous output, and b_i, b_f, b_o and b_c are bias vectors.

In this study, we use the LSTM model defined above to model the sequence of queries issued by a user to accomplish a task. The input x_t in our model is an intent-embedding of a user's queries. In Section 6.4.2, we describe in detail how we acquire

these embeddings.

In our model, the embedding layer is connected to a block of LSTM units. To prevent over-fitting problem, we have used a dropout at the LSTM layer, which randomly drops units and their connections to avoid unit co-adapting [90, 152]. Following previous work, we have used $p = 0.5$ in our dropout network as this value has been reported as a close to optimal value for a wide range of networks in different applications [152]. The output of the last time step in the LSTM feed to a standard feed-forward neural network that contains a single output neuron that uses the sigmoid activation function.

In the learning phase, the derivatives of the loss function are backpropagated through the neural network. Our neural network is trained using the stochastic gradient descent (SGD) algorithm with mini batches, which is widely used algorithm for training neural networks. In order to do hyper-parameter optimization in the learning phase, we have used random search [18], which has been reported to be as good as or better than the grid search in hyper-parameter optimization of neural networks [18].

The random search has been done using a continuous parameter space in range of $[0.0001, 0.1]$ for the learning rate. The chosen learning rates by the random search are adjusted based on the Adam optimization algorithm [108].

6.4.2 Query Representation Learning

We presented our LSTM-based model for user satisfaction prediction in the previous section. We mentioned that the inputs to our model were embeddings that represented the query. In this section, we describe two different representations. One representation is based on an Intent Sensitive Word Embedding (ISWE) while the other is based on unsupervised intent embeddings. Specifically, we answer the research question: *How to model intent-sensitive query representations for user satisfaction prediction?*

Intent-Sensitive Word Embeddings

To learn query representations based on query terms, we explored different word representation models. The word2vec Skip-gram model is one of the state of the art approaches to learn vector representations of words. Word embeddings trained using the Skip-gram model have been shown to be very useful in many tasks [47, 53]. However, in many of the previous efforts, embeddings were generated without taking into consideration the targeted task leading to generic embeddings that might not serve the task well.

For example, Skip-gram model leads to word representations considering "Stop" and "Start" words as being similar. However, although "Stop" and "Start" might be similar based on linear neighbor word context in sessions (e.g., "**start** my jazz playlist on Spotify!" and "**stop** my jazz playlist on Spotify!"), they lead to different queries having completely different intents.

In the rest of this section, we explain our proposed intent sensitive word embedding approach. In our approach, we augment the standard Skip-gram model word embeddings with the query intent information to avoid learning similar representations for words who have similar linear context but different intents. This way, we generate more effective and task oriented word representations compared to the original Skip-gram

word embeddings. The main idea is to add more information when the immediate linear context of the word is not very informative. We train word embeddings based on our proposed Intent-Sensitive Skip-gram model for a query $q \in Q$, having intent $\imath \in I$, and containing a sequence of words $w_1, w_2, ..., w_T$. The objective of the Intent-Sensitive word embedding Skip-gram model is to maximize the log-likelihood of context word-intent pair $w^{\imath}_{t+j}$ given the target word w_t:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-b\leqslant j\leqslant b, j\neq 0}\log p(w^{\imath}_{t+j}|w_t),$$

in which, b is the size of training context ($b = 10$ in all of our experiments), and $\imath$ is intent of the query in which the word occurred.

In order to train the Intent Sensitive Word Embeddings (ISWEs), we have collected intent labels of about 900K IA queries. To collect queries intent labels, we use an in-house micro-tasking platform that outsources crowdwork to judges who regularly perform intent labeling tasks. We presented one query to the judges at a time and they select intent of the given query from a predefined set of intents. The annotations include queries having 266 unique intents like "create_calendar_entry" and "make_call" from 30 different usage domains in IA such as "calendar" and "communication".

We also use negative sampling as discussed in [122]. In negative sampling, having a dataset D of observed $(w, c^{\imath})$ pairs of word w and intent-sensitive context $c^{\imath}$, we generate the set D' including random $(w, c^{\imath})$ pairs assuming they are incorrect. The probability of a $(w, c^{\imath})$ coming from the data is denoted by $p(D = 1|w, c^{\imath})$ and $p(D = 0|w, c^{\imath}) = 1 - p(D = 1|w, c^{\imath})$ is the probability of $(w, c^{\imath})$ coming from the negative examples. Ideally, the $p(D = 1|w, c^{\imath})$ must be high for the word and context pairs observed in the data and low for the random negative samples. The negative sampling training objective is as follows:

$$argmax_{v_w, v_{c^{\imath}}}\left(\sum_{(w,c^{\imath})\in D}\log\sigma(v_{c^{\imath}}\cdot v_w) + \sum_{(w,c^{\imath})\in D'}\log\sigma(-v_{c^{\imath}}\cdot v_w)\right),$$

where $\sigma(x) = 1/(1 + e^x)$, v_w and $v_{c^{\imath}}$ are d-dimensional vectors which are model parameters to be learned using stochastic-gradient updates over the whole corpus including both observed and negative sampled word and context pairs (i.e., $D \cup D'$). In all of our experiments in this chapter, we set $d = 100$. To create the negative sample, we follow Mikolov et al. [122] in creating n negative samples $(w, c^{\imath}_1)$, $(w, c^{\imath}_2)$, ..., $(w, c^{\imath}_n)$ where each $c^{\imath}_j$ sampled based on its unigram distribution raised to the 3/4 power.

According to experiments detailed in [122], n in the range of 5-20 are useful for small training datasets. Thus, following other successful Skip-gram with negative sampling experiments [114], we have chosen 15 as the negative sample size for each positive observed sample in our dataset.

The above objective optimization leads to word and intent-sensitive context pairs having similar embeddings for the pairs observed in the data while scattering negative sampled pairs. In ISWE, words appearing in a similar intent-sensitive context (i.e., context-word and query-intent) should have similar embeddings. We feed a sequence of

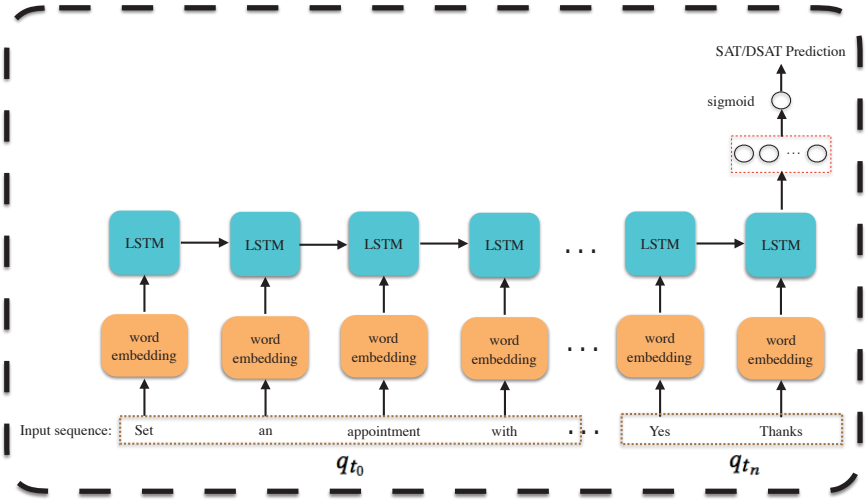


Figure 6.2: User satisfaction prediction model based on sequence of query terms as input.

queries, in which each query contains a sequence of ISWE query terms, to the proposed task satisfaction prediction model discussed in Section 6.4.1. Figure 6.2 shows an example of how we feed ISWE query term representations to the satisfaction prediction network. Effectiveness of the learned query representation in predicting user satisfaction is discussed in Section 6.5.

Unsupervised Intent Embedding

In this model, we choose to forgo the individual words of a query and instead choose to represent the entire query by its intent. To model intent embeddings, we propose using the Skip-gram model [122], which is common for learning word embeddings in NLP [53, 114, 130]. In our proposed Intent2Vec Skip-gram model, each intent $\imath \in I$ is associated with a vector $v_i \in R^d$, where I is the intent vocabulary, and d is the embedding dimension. In all of our experiments, we set $d = 100$. The training objective of the Intent2Vec model is to find intent representation, which are effective to predict the intent of surrounding queries in a task or a session. Formally, given a sequence of intents $\imath_1, \imath_2, ..., \imath_T$ in a session, the objective of the Intent2Vec Skip-gram model is to maximize the following function:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-b\leqslant j\leqslant b, j\neq 0} \log p(\imath_{t+j}|\imath_t),$$

in which, b is the size of training context, which is set to $b = 10$ in all of the experiments in this chapter. To train the intent embedding, we have used negative sampling as presented in Section 6.4.2 and chose 15 as the number of negative samples for each positive sample observed in our dataset.

In order to train query representations based on the Intent2Vec model, we sampled about 500K IA queries issued in a 3 month period of a commercial smart speaker usage.

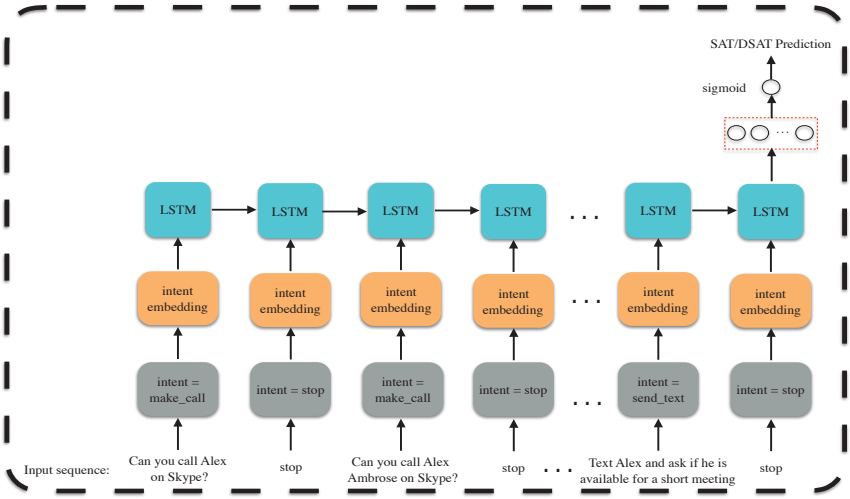


Figure 6.3: User satisfaction prediction model based on sequence of query intents as input.

This dataset does not have any relation with the 900K queries dataset used for training ISWE. We use a high-quality production-level offline intent classifier to assign intents to each query. Our query sample has 181 unique intents like "check_weather" and "volume_down" from 23 different domains such as "weather" and "media-control". As previously mentioned, Intent2Vec trains intent embeddings based on neighboring query intents. Thus, to train intent embeddings, Intent2Vec requires a sequence of queries issued in each session as an input. In order to create a sequence of query intents for each session, we need to identify sessions from the raw IA log data. To do this we follow the approach of [80] to identify session boundaries, which leads to creation of about 69K sessions based on the one hour session boundary identified using the approach presented in [80]. Using sessions as input of I2V leads to training intent embeddings based on a larger context size compared to training the intent embeddings on tasks. Training the intent embeddings on sessions provides representations that considers both within task and cross-task contexts in IA sessions.

To predict task satisfaction, we feed a sequence of query intent representations to the model discussed in Section 6.4.1. Figure 6.3 shows an example of how we feed a sequence of queries to the user satisfaction prediction network. We first assign an intent to each user query using a high-quality production-level offline intent classifier. We then feed a sequence of intent embeddings query representations to the user satisfaction prediction network. In our proposed user satisfaction prediction models, we just consider users' queries as input and we do not use system responses, which are directly controlled by the IA, as input and feature to the user satisfaction prediction model. By excluding system responses, we avoid using endogenous features (i.e., features that the search engine has control over [167]) in our online user satisfaction prediction model. Evaluation of the model is available in Section 6.5.

In this section, we have defined the task satisfaction problem and our proposed model to address the problem. We have also described the intent-sensitive word representation

learning and Intent2Vec query-intent representation learning models with an aim of learning effective query representations for task satisfaction prediction. In the next section, we present a set of experiments evaluating these models.

6.5 Experimental Evaluation

In this section, we evaluate our proposed models by answering the research question: *How effective is the proposed intent-sensitive user satisfaction model in evaluating intelligent assistants on smart speakers?*

6.5.1 User Satisfaction Judgment Crowdsourcing

This study is based on a random sample of users interaction logs with a commercial IA being used on a smart speaker during August, September and October, 2017. Our random sample includes user sessions having one to ten queries. As it is not easy to collect explicit feedback about actual users' satisfaction, crowdsourced judgments, which have been widely used to obtain labeled data for different problems including user satisfaction in IAs [119, 169], is used. To collect user satisfaction judgments, we use an in-house micro-tasking platform that outsources crowdwork to judges who regularly perform relevance judgment tasks. We removed all the personal identifiable information (PII) from the sessions before sending them for judgment.

A detailed guideline including a video explaining how to judge user satisfaction was shown to the judges. We presented a whole session to judges, and asked them to assess query-level satisfaction and session level satisfaction of a user in the given session. We also collected task identification labels for a user session, in which crowdsource workers judged whether the user was trying to fulfill the same information need as the previous query by issuing the current query. To judge user satisfaction and task identification, judges could read or listen to the user's query, read system response, look at the original timestamp of queries, and read or listen to the user's previous or future queries.

In order to obtain a high-quality user satisfaction labels, at least 3 and at most 5 crowdsource workers judged each session. Qualifying tests and spam detection were used to filter out low-quality judgments. The final label used for satisfaction is based on a majority vote. We randomly sampled over 1700 user sessions in the smart speaker IA and collected user satisfaction labels for them. We measure inter-rater agreement using Fleiss' Kappa [54].

The goal of this study is to measure task satisfaction. To create tasks, we used the majority vote as task boundary labels. Specifically, we create tasks based on sequence of queries issued by a user up to a point that the task is ended using the task boundary collected labels. In our collected dataset, the crowdsourced task boundary labels led to 3105 tasks including 6920 query-level satisfaction labels. The Kappa value is 0.42 for the collected query-level user satisfaction labels and 0.72 for the collected task boundary labels.

After creating the tasks, we need to assign a label to each task. In Figure 6.4, we have shown the correlation between percentage of satisfactory queries in a session and the session satisfaction probability based on the crowdsourced session satisfaction

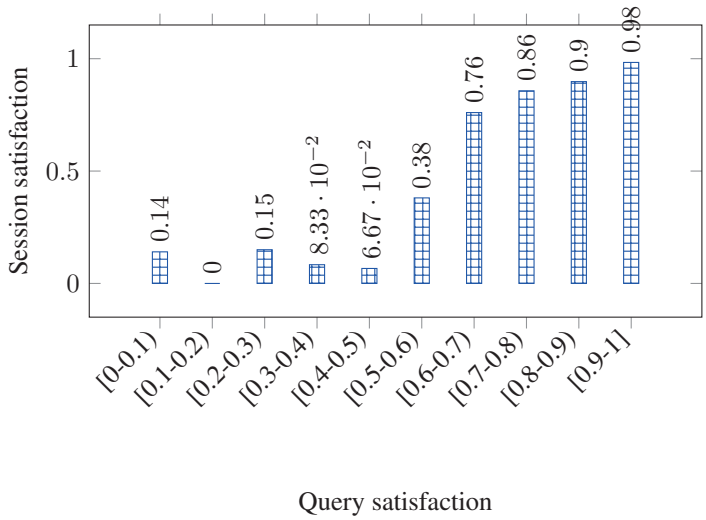


Figure 6.4: Impact of percentage of satisfactory queries in a session on session satisfaction. This chart is based on a crowdsourced judgments collected in Section 6.5.1.

collected labels. According to our observation on session-level user satisfaction in Figure 6.4, having from 60% to 70% user query-level satisfaction in a session leads to session-level satisfaction in 76% of the cases. Overall, having at least 60% user query-level satisfaction leads to 93% session-level satisfaction compared to just 19% user satisfaction for tasks having less than 60% user query-level satisfaction. Therefore, we have intuitively used the 60% threshold for task-level satisfaction labels, which means that if 60% or more of queries issued in a task are labeled SAT, then we label the task as SAT.

6.5.2 Baselines

We consider three baselines when evaluating our models.

1. Query reformulation (QR) [86] is one of the defined baselines, which classifies a task as DSAT if the last query of a user in the task is a reformulation of the second to last query with no user interaction after the last query. Otherwise, it classifies the task as a SAT experience. We use the method of [86] to determine if a query was reformulated.

2. The second baseline is a variant of the generative model (GM) [83], which uses a sequence of query terms to predict satisfaction. The GM is a mixture model composed of SAT and DSAT components. Give a sequence of interactions, the goal during classification is to identify whether the sequence was generated by the SAT or DSAT components of the mixture model [83]. The model was originally used to predict search success based on a sequence of actions in a session including clicks, query reformulation and queries. We use the query terms

Table 6.2: Nearest neighbors of three examples in different representation spaces based on cosine similarity. Both word2vec skip-gram baseline and the intent-sensitive word embeddings are trained on a same dataset.

Word	good
Word2Vec Skip-gram	nice, great, wonderful, **bad**, lovely
Intent-Sensitive WE	nice, great, wonderful, fine, decent
Word	yes
Word2Vec Skip-gram	yeah, **not**, okay, **no**, ok
Intent-Sensitive WE	yeah, yep, sounds, correct, ok
Word	cancel
Word2Vec Skip-gram	delete, remove, erase, **edit**, **set**
Intent-Sensitive WE	remove, delete, disable, dismiss, clear

as sequence of actions in the GM as they are the only means of user interactions in smart speaker IAs.

3. In order to evaluate effectiveness of our proposed ISWE in user satisfaction predictions (ISWE-LSTM), we have defined another baseline based on our proposed user satisfaction prediction model detailed in Section 6.4, yet using original Skip-gram representations of query terms as input to the network (W2V-LSTM).

6.5.3 Experimental Result

We conduct experiments to evaluate the effectiveness of the proposed ISWE in capturing word-intent and the intent-sensitive user satisfaction models in differentiating between SAT and DSAT tasks. We also measure the impact of task-type on user satisfaction and the effectiveness of the proposed models in predicting satisfaction in tasks having different types. Specifically, we address the following research questions in this section:

1. *How effective are the intent-sensitive word embeddings in estimating word similarity by capturing word-intent compared to the Skip-gram word2vec model?*

2. *How effective is the intent-sensitive user satisfaction model compared to the user satisfaction prediction baselines?*

3. *How effective is the intent-sensitive user satisfaction model in different task types?*

Intent-Sensitive Word Embeddings in Word Similarity

The word representations learned by our proposed ISWE are expected to capture word intent. Table 6.2 shows three different examples of the nearest neighbors of words based

Table 6.3: Task satisfaction prediction result. * indicates statistical significant improvements based on Student's paired t-test and Wilcoxon signed-rank test ($\rho < 0.05$).

Classifier	P	R	Acc.	F1
QR	0.5117	0.5059	0.6248	0.5086
GM	0.6056	0.6068	0.6345	0.6062
W2V-LSTM	0.6513	0.6209	0.6910	0.6356
I2V-LSTM	0.6254	0.5528	0.6566	0.5862
ISWE-LSTM	**0.6891***	**0.6669***	**0.7174***	**0.6778***
Impr. over GM (%)	13.80%	9.90%	13.07%	11.81%
Impr. over W2V-LSTM (%)	5.80%	7.41%	3.82%	6.63%

on cosine similarity. As it is shown in Table 6.2, in contrast to the Skip-gram word2vec embeddings that is not capable of capturing word-intent by putting words like "yes" and "no" very close in the representation space, ISWE can effectively capture word-intent by scattering words having different intents. For example, Table 6.2 indicates that the top-5 similar words to word "yes" based on cosine similarity of word2vec Skip-gram word representations are "yeah", "not", "okay", "no" and "ok", which includes words with very different query-intents. On the other hand, the top-5 similar words to word "yes" based on cosine similarity of ISWE word representations are "yeah", "yep", "sounds", "correct" and "ok", which is an example of how ISWE captures word-intent in the word representations.

Capturing word-intent in the word representations can be very beneficial for user satisfaction prediction, as understanding user query intent incorrectly can lead to a DSAT experience. For example IA might proactively ask a user "You have a reminder. Should I read it?". Then, if the user says "no" and the IA starts reading the reminder, the user would have a DSAT experience. However, if a user satisfaction prediction model cannot capture the user's intent, then it might classify the task to the SAT category as it considers words "yes" and "no" as very similar words.

ISWE performs better than the standard skip-gram word embeddings because the objective function of the ISWE does not just depend on words linear context in sentences. In fact, query-intent plays a major role in the objective function, which scatters words having different intents. In the next part, we discuss effectiveness of our proposed intent-sensitive user satisfaction model compared to the baselines.

Impact of Intent Sensitive Word Embeddings on User Satisfaction

We now answer our research question: *How effective is the intent-sensitive user satisfaction model compared to the user satisfaction prediction baselines?*

Table 6.3 shows the user satisfaction prediction results of the proposed intent-sensitive user satisfaction prediction models compared to the baselines. In these experiments, the classification threshold is 0.5. The experiment is based on a 5-fold cross validation, where three folds were used for training, one for validation, and one for testing. We repeat the process for all the five folds and report the average of the

Table 6.4: Satisfactory session distribution over different task type.

Task Type	Distribution	Sat Task Percentage
Single-Query	54.33%	60.05 %
Two-Turn	18.20%	63.00 %
Multi-Turn	27.47%	77.02 %

evaluation metrics. According to our experimental results, the query reformulation baseline performs poorly in the prediction of user satisfaction. One possible explanation for this is the dialogue nature of queries, where users might refine their queries to give more details about their information needs. It also has a huge bias toward predicting the SAT class, which could be the second possible explanation of its poor performance. On the other hand, the generative model performs much better than the query reformulation baseline, which provides a good baseline for our proposed model evaluation.

As it is shown in Table 6.3, the intent-sensitive user satisfaction based on ISWE query representations leads to a significant improvement over all the baselines in terms of all the defined evaluation metrics. Specifically, it improves task satisfaction prediction over the generative model from 0.6062 to 0.6778 in terms of $F1$ metric. The proposed model also has a statistically significant improvement over the task prediction based on the Skip-gram query representation, in which the $F1$ metric is improved from 0.6356 to 0.6778.

One possible explanation of the improvements achieved by the ISWE-LSTM over the I2V-LSTM is that in contrast to the I2V-LSTM model getting sequence of queries as an input, the ISWE-LSTM gets sequence of intent-sensitive query terms as an input. Therefore, effectiveness of the ISWE-LSTM is less affected by single-query tasks. Furthermore, ISWE-LSTM has improvements over other baselines that also use query terms as input, because the ISWE model takes advantage of the query intents for satisfaction prediction.

Our experimental results shows that the user satisfaction prediction based on Intent2Vec query-intent representations (I2V-LSTM) does not lead to an improvement over the generative model in terms of the F1-measure. However, as it is shown in Table 6.4, 54.33% of tasks in our crowdsourced dataset contain a single query, which is not ideal for training the user satisfaction prediction based on a sequence of intents. We suspect that the I2V-LSTM would do better for long tasks having multiple queries compared to single query tasks. We investigate this in the next experiment.

Impact of Task Type on User Satisfaction

In our final experiment, we answer our research question: *How effective is the intent-sensitive user satisfaction model in different task types?*

The dialogue nature of user queries in IAs leads to tasks, in which users might issue multiple queries to accomplish a single information need. However, although users are capable of having long conversation with the IA, the majority of tasks are single query tasks in smart speakers [80]. In Table 6.4, we categorized tasks into three different types: 1) Single-Query tasks are tasks having a single query and a single system response, 2) Two-Turn tasks are those tasks where the user issues a second query after receiving a

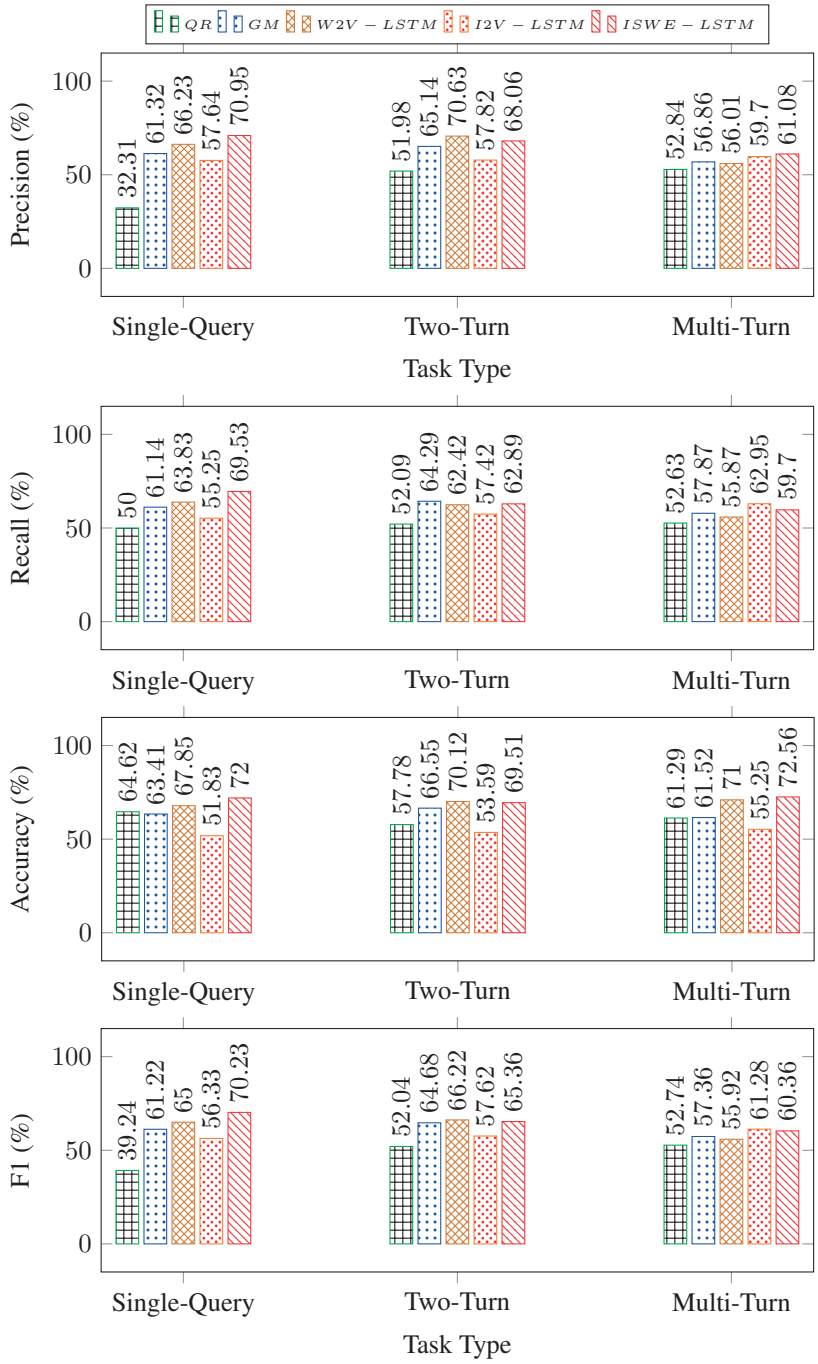


Figure 6.5: Effectiveness of intent-sensitive user satisfaction models compared to the baselines in different task types.

system response to their first query, and 3) Multi-turn tasks are tasks having more than two user queries (i.e., multiple turns).

According to Table 6.4, users seem more satisfied in tasks having multiple queries. One possible explanation of such a behavior is that the IA retains the context of the dialogue, and shorter sessions are more probable to be sessions that IA was not able to retain the dialogue context and consequently the user abandoned the task.

As the user satisfaction level varies based on the defined task-types, and user behavior might be different in each of them, we investigate user satisfaction prediction for each task-type separately. Figure 6.5 details our proposed intent-sensitive user satisfaction prediction models effectiveness compared to the baselines in different task types. As expected, the query reformulation approach does not work well for the single query tasks as it assign SAT labels to all tasks. The query reformulation does not work well in two-turn and multi-turn tasks either, which shows that query reformulation is a poor approach for measuring user satisfaction in IAs on smart speakers. The generative model performs better for two-turn tasks compared to multi-turn tasks. We observed similar results for W2V-LSTM model in terms of precision and F1.

Figure 6.5 shows that our proposed intent-sensitive models perform better than all the baselines in single-query and multi-turn task-types in terms of all common metrics; however, the effectiveness of the generative model, W2V-LSTM and ISWE-LSTM in user satisfaction prediction in two-turn sessions are similar. In fact, the generative model and W2V-LSTM perform slightly better than ISWE-LSTM. The generative model performs better than W2V-LSTM and ISWE-LSTM in terms of recall of two-turn tasks, and W2V-LSTM is better than generative model and ISWE-LSTM in terms of precision. However, in terms of F1 as a more fair metric to evaluate classification, the genarative model, W2V-LSTM and ISWE-LSTM perform very similar.

Furthermore, as it is shown in Figure 6.5, the I2V-LSTM model improves as task length increases. As expected, the I2V-LSTM improves the task satisfaction prediction based on F1 score from 56.33 in single-query tasks to 61.28 in multi-turn tasks. Specifically, in single-query tasks, the I2V-LSTM performs poorly because of using a single query-intent as input to the sequence classification. In two-turn tasks, I2V-LSTM performance is improved as it uses a sequence of 2 intent representations in contrast to a single intent representation in single-query tasks. Using longer sequence of intents leads to a better understanding and prediction of user satisfaction in I2V-LSTM model, which makes the I2V-LSTM model the best performing user satisfaction model for multi-turn tasks in terms of F1.

To summarize, in this section, we have presented experimental results and shown effectiveness of ISWE compared to the original Skip-gram word2vec in capturing word-intents and learning query representation for user satisfaction prediction. Then, we discussed satisfaction in different task-types. In contrast to two-turn tasks, in which the generative model, W2V-LSTM and ISWE-LSTM models perform similarly, our experimental results indicates that the I2V-LSTM intent-sensitive user satisfaction prediction model is the best performing system in terms of F1, which is our main evaluation metric. Moreover, due to using intent-sensitive word-level representations of queries, ISWE-LSTM intent-sensitive user satisfaction prediction model is the best approach for single-query tasks.

6.6 Conclusion

In this chapter, we investigated the user satisfaction prediction problem in intelligent assistants (IAs) on smart speakers, in which the only means of user interactions with the IA is a sequence of user queries. Our main research question was: *How to evaluate user satisfaction in Intelligent Assistants based on user queries?* To do this, we proposed a new implicit signal from user queries to predict user satisfaction, which is query intent. Using the query intent, we proposed an intent-sensitive user satisfaction prediction model. Learning effective query representations as input of the user satisfaction prediction model is one of the main contributions of this chapter. To train intent-sensitive query representations, we proposed two intent-sensitive models.

We first proposed an intent-sensitive word embeddings (ISWE) learning model, which is a modification of the popular word2vec Skip-gram model. According to our experimental results, the ISWE are very effective in capturing query term intents compared to the original Skip-gram model. Second, we proposed an unsupervised Intent2Vec Skip-gram model to capture linear context of query intents in user sessions.

According to our experiments, incorporating ISWE as the input to a user satisfaction prediction model based on a sequence of query terms leads to a statistically significant improvement over all the defined baselines. Furthermore, we further evaluated the effectiveness of the proposed intent-sensitive user satisfaction prediction models in different task types and showed that the proposed intent-sensitive user satisfaction model based on ISWE performs better than the baselines in single-query and multi-turn task-types, yet performs similar to baselines in two-turn task-type in terms of the common classification metrics. One possible explanation for this could be the nature of queries used in the training phase of ISWEs. The training set for learning the ISWEs was query and intent pairs without considering the whole task context. Therefore, the learned ISWEs are optimized for a single-query level context, and consequently the user satisfaction prediction model based on ISWE performs better in single-query task type. Furthermore, our experimental results indicate that compared to the two-turn task type, our proposed user satisfaction prediction based on ISWE performs better than baselines in multi-turn task type as it gets more inputs about the task-context by getting more queries as input. In addition, the experimental results indicate that the number of queries in a task (task length) has a positive impact on the user satisfaction prediction based on a sequence of Intent2Vec query intent representations, which leads to the best performing system in terms of F1 in multi-turn task-type.

We have not studied more advance neural network architectures, as it is not the focus of this chapter and we leave this to future work. Furthermore, other query features could be added to the intent-sensitive user satisfaction prediction model with the aim of improving the satisfaction measurement results. We also leave this for future work.

7 Discussions and Conclusions

In this chapter, we first revisit our research questions introduced in Chapter 1 and summarize main findings and implications of our research in Section 7.1. Then, in Section 7.2, we describe the main results and limitations of our work and the possible future directions.

7.1 Main Findings

The main aim of this thesis is to study *how to model users interacting with smart devices to improve their experience in the physical space?* To study our main aim, we first focused on modeling users interacting with smart environments to improve user experience in physical spaces such as a smart museum. We then investigated creating and maintaining a reusable test collection for evaluation and performance improvement of the contextual suggestion systems to improve user experience. At last, we improved user experience in performing tasks with their smart speaker IAs at their smart homes by identifying tasks and sessions, and then predicting user satisfaction while performing a task.

In this section, we revisit the research questions and summarize our findings regarding each research question.

7.1.1 POI Recommendation in Smart Environments

We started with the task of onsite POI recommendation in a smart environment with an integrated Internet of Things (IoT) and asked:

RQ1 How to model users' information interaction behavior with IoT having an aim of providing a personalized onsite POI recommendation?

To answer this question, we tracked users' onsite physical interactions in a smart museum by logging their check-in behavior. In addition, by logging users click-through behavior, we tracked users' online digital interactions at the smart museum website while using the museum's search engine to explore their collection. We then trained a model to learn from users' interaction behavior to predict a POI that a user is interested in and going to interact with. In particular, the behavioral user model predicts relevant unseen POIs to the given user and context based on the user's interaction behaviors in

the context. We cast the context-aware recommendation problem to a ranking problem, in which the relevance probability of POIs to the user and context pairs were used to rank the unseen POIs.

We defined three types of users' interaction with IoT, namely, explicit context given by the user while interacting with smart devices, onsite physical interaction with smart devices, and online digital interaction with the objects being shown in the smart devices. We then studied the strength of using each type of user interaction behavior with IoT in understanding users' onsite information interaction preferences. The experimental result showed that the POI recommendation system trained using features extracted from a combination of both onsite physical and online digital information interaction behaviors (i.e., online features) performs better than the ones trained by explicitly given context or onsite information interaction behavior. Therefore, we conclude that there is a similarity between onsite physical and online digital interaction preferences that causes an improvement in the onsite POI recommendation effectiveness in the smart museum.

We further studied the critical one-shot POI recommendation problem. According to our experimental results, learning a deep multilayer perceptron based on features extracted by online interaction behaviors led to a significant improvement over the hard-to-beat defined baselines in terms of all the defined evaluation metrics. Specifically, it had a statistically significant improvement over the best defined baseline with a 23% improvement in terms of $p@1$ and 11% improvement in terms of MRR. Thus, our proposed model trained using online features is very effective in the critical one-shot onsite POI recommendation in the smart museum.

Furthermore, we analyzed the impact of seen objects set size on the performance of the onsite POI recommendations. Our experimental result showed that recommendation performance is generally increased proportionally by the seen object set size. However, due to external factors such as onsite interaction biases, the proposed model based on just onsite physical interaction behavior is more sensitive to seen set size compared to the proposed model based on online digital interaction behavior. In addition, our proposed deep MLP model based on online features is more robust in different seen set sizes and performs better than other models and baselines at all seen objects set sizes.

Our general conclusion of this chapter is that it is possible to combine information interactions in the online digital and onsite physical world for effective onsite POI recommendation in smart environments, thereby effectively blending real-world and online behavior in principled ways. As data of user interactions with IoT is typically far more sparse than users' online interaction data due to physical or geographical constraints on users requiring to be physically in the smart space, this line of research is an attractive direction for both academia and industry.

7.1.2 Test Collection Building for Contextual POI Recommender Systems

Creating effective POI recommender systems and constantly improving their performance in smart environments requires a reusable test collection. To create a reusable test collection for the evaluation of personalized context-aware recommender systems in smart environments such as smart cities, we organized TREC 2016 contextual sug-

gestion track and answered our second research question:

RQ2 How to create a reusable test collection for the Contextual Suggestion problem?

To create a reusable test collection for the contextual suggestion, which has been proved to be a difficult task due to the dynamic nature of the collection and personalization effect on the pool depth [72, 74], we first released the TREC CS web corpus. The TREC CS web corpus is a crawl of the TREC contextual suggestion test collection. To overcome the dynamic nature of the contextual suggestion collection and separate this effect from the personalization effects, we fixed the TREC 2016 contextual suggestion test collection's content by releasing the TREC CS web corpus. We then used a multi-depth pooling approach to improves the reliability of the contextual suggestion systems scores based on measures at ranks deeper than the traditional pool cut-off.

Our experimental results indicated that the fraction of judged documents gently decreases after the hard pool cut-off in the TREC 2016 contextual suggestion test collection. This result is an improvement over the stability of the fraction of judged documents in the TREC 2015 contextual suggestion test collection, in which the fraction of judged documents dropped dramatically after the pool cut-off.

Furthermore, Leave-One-Team-Out (LOTO) reusability test showed that the TREC 2016 contextual suggestion test collection should be used with some care based on the P@5 metric. However, the test collection appears to be reusable based on the more stable evaluation measures for incomplete test collections. Specifically, the test collection has got a perfect system ranking correlation between official TREC system ranking and the LOTO system ranking based on the Kendall's τ using statistical significant inversions using MAP and bpref metrics.

At last, we created a fixed test collection and cast the contextual suggestion task from ranking to a reranking problem in Phase 2 of the experiments. The Phase 2 experiment is a reranking task, in which a set of suggestion candidates is provided for each request. Thus, we have all the judgments of the suggestions available in the suggestion candidates, which facilitates the reuse of the contextual suggestion test collection by design. However, the TREC contextual suggestion ranking test collection is not reusable based on early precision-based metrics such as P@5, and the test collection reusability may degrade due to the dynamic nature of the test collection.

In the next chapter, we discuss our approach for maintaining and improving the reusability of personalized ranking test collections. Specifically, the next chapter focuses on maintaining the test collection created for contextual suggestion ranking in this chapter.

7.1.3 Test Collection Maintenance in Dynamic Domains

Although we have achieved the creation of a reusable test collection for contextual POI ranking using MAP and bpref metrics, the created test collection is built based on open web pages that make maintaining the reusability of the test collection challenging due to the dynamic nature of the collection. As the reusability of the created test collection for the contextual POI recommendation may degrade in the future, we studied a corpus and a test collection expansion approach to maintain and improve the reusability of

test collections in the dynamic domain. In this study, we answered our third research question:

RQ3 Can we build a reusable test collection for a dynamic domain by injecting judged documents into a test collection with sparse judgments?

To answer the research question, we first analyzed reusability of the open web-based and ClueWeb12-based TREC 2014 contextual suggestion track test collections built by following a same experimental design. Our findings indicated that both open Web-based and ClueWeb12-based TREC 2014 contextual suggestion track test collections are not reusable based on Leave Out Uniques (LOU) experiments and system rank correlation metrics. Our experimental results indicated that shallow pool depth and personalized nature of the problem contributed negatively to the reusability of both of the test collections.

We then experimented with the ClueWeb12-based test collection expansion using Web pages available in the Open Web test collection to improve the reusability of the ClueWeb12-based test collection. Our approach was simply adding judged pages of the open Web test collection to the existing ClueWeb12-based test collection and corpus. We then had retrieval systems to retrieve those relevant judged open Web pages from the expanded corpus. Our experimental results showed that the retrieval systems were able to retrieve a fair and stable fraction of the added relevant open Web pages, which led to a more consistent system ranking based on the contextual suggestion evaluation metrics and LOU test.

Specifically, the LOU reusability test showed a perfect system rank agreement over a set of nine retrieval systems. Furthermore, we found that the fraction of judged pages from the retrieved pages by the nine retrieval systems improved significantly compared to the original ClueWeb12-based test collection, and the fraction of judged pages decreased gently over the retrieved page ranking.

As the nine retrieval models were not pooled runs, the LOU test is not fully evaluating the pooling bias of the test collection. In order to simulate the pooling effect of a set of non-pooled runs in reusability of a test collection, we proposed a variant of the LOU test which leaves uniques contribution of either a team or a run in the test collection (i.e., Leave One Team In (LOTI) and Leave One Run In (LORI)). The critical LORI and LOTI tests indicated that the expanded test collection is reusable for evaluation of personalized systems using the stable MAP and bpref metrics but not based on the P@5 which is known to be less stable due to being an early precision metric.

Our further investigation on reusability of the expanded test collection using the critical LOTI and LORI tests using non-personalized retrieval systems showed that the expanded test collection is reusable based on all the selected common metrics including P@5, which is an early precision-based metric. This experimental result emphasized once more on the challenges of test collection building for personalization tasks with shallow pooling.

Our general conclusion in this chapter is that our proposed test collection expansion approach is effective for maintenance and improving the reusability of test collections for offline evaluation in dynamic domains such as the Web.

In the previous chapters, to improve user experience in physical spaces such as smart

museums and cities, we developed an effective contextual POI recommender system in smart environments, created a reusable test collection for evaluation of contextual POI recommendation, and proposed an approach to maintain reusability of contextual POI recommendation test collections. In the next two chapters, we focus on improving user experience in interacting with smart speaker IAs at physical spaces such as smart homes.

7.1.4 Task and Session Identification on Smart Speakers

To improve user experience on smart speaker IAs, we first need to segment smart speaker IAs interaction logs into tasks and sessions, which is a key element of online evaluation. Thus, we focused on modeling user interaction behavior on smart speakers to first identify tasks and sessions from user interaction raw logs and then use the identified tasks as inputs of a user satisfaction prediction model to measure task-level user satisfaction on smart speaker IAs.

To identify tasks and sessions, we first experimented with an application of a 3-component Gaussian Mixture Model (GMM) to fit user inter-activity times on smart speakers to jointly identifying task and session boundary cut-offs based on smart speaker IA interaction logs. We further investigated the impact of learning phase and usage domain on task and session boundary cut-offs based on user interaction logs of two different IAs being used in smart speakers to identify task and sessions more accurately and answer our fourth research question:

RQ4 What is the impact of the learning curve and task domain on task and session boundaries when interacting with intelligent assistants?

To answer the research question, we first experimented with 2- and 3-components GMM to estimate task and session boundary cut-offs on smart speaker IAs. Our experimental results indicated that GMM is an effective model to identify tasks and sessions in smart speaker IAs. Furthermore, based on the experimental results, the 3-components GMM estimates task boundary cut-offs on smart speaker IAs better than the 2-components GMM. However, when we had a deeper analysis on the impact of contextual factors on task and session boundaries, we observed different user interaction behavior in different context on smart speakers.

Learning phase was one of the contextual factors studied in this part of the thesis. Our experimental results showed that user inter-activity times in the learning phase of the smart speaker usage has 2 main clusters of inter-activity times, which means 2-component GMM is a more rational model for task or session identification compared to the 3-component GMM in the learning phase. However, our experimental results indicated that a new cluster of inter-activity times appeared after the learning phase (normal phase) of the smart speaker usage, which made 3-component GMM a more rational choice for task and session identification compared to the 2-component GMM. Our experimental results led to a conclusion that while using 2-component GMM is a reasonable fit of users inter-activity times in their learning phase, fitting a 3-component GMM is more effective during the normal-phase.

In addition to the learning phase, we did experiments on impact of the task domain on task and session boundaries. The experimental results showed that task and session

boundaries differ across domains. Thus, our general conclusion of this chapter is that task and session boundary cut-offs are not static and they are dependent to contextual factors such as learning phase and domain of the usage.

In the next chapter, the identified tasks from the smart speaker IAs interaction logs is used as input of a user satisfaction prediction model for improving user experience with smart speaker IAs at smart home physical spaces.

7.1.5 User Satisfaction Prediction on Smart Speakers

Finally, we focused on modeling users interaction behavior on smart speakers to predict their satisfaction while performing a task with smart speaker IAs, which is a key element of improving smart speaker IA performance and improving users experience at a smart home physical space. IA tasks used in experiments of this chapter are identified by the task identification approach presented in the previous chapter. We studied user satisfaction prediction on smart speaker IAs, in which the only means of user interactions with the IA is a sequence of user utterances, and answered our last research question:

RQ5 How to evaluate user satisfaction in Intelligent Assistants based on user queries?

To answer the last research question, as user utterance is the only means of user interaction with smart speakers, we did extensive experiments on training effective query representation to be used in user satisfaction prediction models. According to the experimental results, our proposed intent sensitive word embeddings (ISWE) learning model are very effective in capturing query term intents compared to the original Skip-gram model. Furthermore, our proposed unsupervised Intent2Vec Skip-gram model captures linear context of query intents in user sessions

Our experimental results indicated that incorporating ISWE as the input to a user satisfaction prediction model based on a sequence of query terms leads to a statistically significant improvement over all the defined baselines. We further evaluated the effectiveness of the proposed intent-sensitive user satisfaction prediction models in different task types and showed that the proposed intent-sensitive user satisfaction model using ISWE query representation performs better than the baselines in single-query and multi-turn task-types, yet performs similar to baselines in two-turn task-type in terms of the common classification metrics. A possible explanation for this experimental result is that ISWE was trained using a set of queries and their intent without considering the corresponding task context. Thus, the learned ISWEs are optimized for a single-query level context. Furthermore, our experimental results showed that compared to the two-turn task type, our proposed user satisfaction prediction based on ISWE performs better than baselines in the multi-turn task type as it gets more inputs about the task-context by getting more queries as input. In addition, the experimental results indicated that the number of queries in a task (task length) has a positive impact on user satisfaction prediction based on a sequence of Intent2Vec query intent representations.

Our general conclusion of this chapter is that query intent is an effective implicit signal from user queries to predict user satisfaction. According to the experimental results, our proposed user satisfaction prediction model using ISWE is the best model for

the single-turn tasks, which are the majority of the tasks on smart speakers. Furthermore, using Intent2Vec query representation in our user satisfaction prediction model led to the best performing user satisfaction model on smart speakers IAs for multi-turn tasks in terms of the F1 evaluation metric.

7.2 Discussion and Future Work

As a general conclusion of the thesis, user experience with smart devices in physical spaces can be improved by user interaction behavior modeling. To this aim, we either reused an already well-known user interaction signal such as click-through interaction or proposed a new user interaction signal such as user intent.

Modeling users interacting with Web search engines has been studied extensively in the literature. However, modeling users interacting with smart environments with an aim of exploring information is relatively less studied and has challenges such as data sparsity and cold start problem in personalization. We found similarity between user online behavior interacting with search engine of a museum and onsite behavior interacting with smart devices in a smart museum. Specifically, modeling users interacting with both onsite IoT sensors and online search engine result pages can improve onsite POI recommender systems and users experience in physical spaces such as smart museums.

Furthermore, creating and maintaining reusable test collections for offline evaluation of personalized contextual suggestion is challenging. Personalization, shallow pool depth and dynamic nature of the test collection contribute negatively in the reusability of the test collection. However, personalization is important aspect of contextual suggestion systems that can have direct impact on user experience. Thus, creating reusable test collections for personalized contextual suggestion systems is crucial for improving performance of contextual suggestion systems and providing better user experience in physical spaces such as smart cities. To create a reusable personalized test collection, we created an archive of the TREC contextual suggestion test collection Open web pages to reduce the negative impact of the dynamic nature of the test collection on their reusability. In addition, we experimented with a test collection expansion approach for maintaining reusability of the test collection. To try fixing the negative impact of a shallow pool depth in traditional top-n pooling approach on their reusability, we proposed an approach to have a cost-effective pooling approach that leads to creation of test collections, which are less sensitive to a defined pool depth.

At last, modeling users interacting with smart speakers IAs leads to understanding the users' behavior better, prediction of their satisfaction while interacting with the smart device and then improving their experience with smart speaker IAs located in a physical space such as their smart home. In particular, different contextual factors such as learning phase and domain of usage have impact on users interaction behavior on smart speakers. Modeling users effectively by considering these contextual factors in task and session identification, which is a key element of task and session level evaluation of smart speakers, leads to a more accurate task identification model and consequently a more effective task-level user satisfaction prediction model. In addition, by modeling users interacting with smart speakers to measure user satisfaction, we effectively predict user satisfaction, which is an important signal to improve user experience on smart

speaker IAs in physical spaces such as smart homes.

Our current understanding of how users interact with emerging smart devices is fragmented at best, and this thesis is one of the first to explore this important topic. Both novel devices and applications are being invented, and users evolved by discovering novel use cases and conventions. Rather than offering definite final answers, this thesis hopes to have done some important initial steps, with every step prompting more research questions to explore.

Although we experimented with real data extracted from production environments in Chapters 2, 5, and 6, one of the limitations of this thesis is not seeing the improvement in user experience based on our proposed models in physical spaces such as smart environments in a production environment. Thus, an interesting line of research for future work would be applying the lessons learned in this thesis into a production environment and observe an actual impact on user experience. In [170], we have incorporated user satisfaction in IAs to improve user experience and providing more effective IA that is capable of performing complex tasks.

In this thesis, we studied user behavioral modeling for unseen POI recommendation in smart museums. As a future work, Modeling users' onsite and online interaction behavior with the aim of route recommendations in a smart museum is a direction of research that can improve users' satisfaction at their museum visits. One of the limitations of using smart devices in smart museums is reducing the social aspects of museum visits [71]. To bring social aspects of museum visits to users' experience at a smart museum, similar people recommendation in the physical space based on onsite physical and online click-through interaction behavior can lead to the creation of groups with similar interests. In this way, museum visitors would enjoy a group visit at a smart museum, in which group members have similar interests, without sharing any personal information.

As it was discussed in detail in chapter 3 and 4 of the thesis, personalization and shallow pool depth contributed negatively to the reusability of the TREC Contextual Suggestion track test collections. We proposed a cost-effective approach for creating reusable test collections for personalized contextual suggestion systems in chapter 3. However, we are in an early stage of tackling reusable personalized test collection creation, and further research is required. Thus, a cost-effective reusable personalized test collection creation remains an important line of research for future work.

In chapter 5 of the thesis, we studied task and session identification on smart speaker IAs. We observed that user behaviors differ on smart speaker IAs compared to IAs on desktop computers or mobile devices. As there is an interest in integrating IAs in different smart devices, including smart cars and watches, task and session identification should be studied on the other smart devices with an integrated IA to improve evaluation of smart device's performance in future works.

Different smart devices provide various means of user interactions. Thus, user behavior may be different in these devices, which makes the applicability of current user satisfaction prediction models on the other smart devices questionable. A line of research for future work is user satisfaction prediction in using IAs on other smart devices such as smart cars.

Furthermore, we have not studied more advance neural network architecture for user

satisfaction prediction in the thesis as it was not the main focus of the research. Trying some other neural network architectures such as attention networks and transformers to improve user satisfaction prediction on smart speakers is left for future work. As we are planning to use more complex neural network architectures in future work, addressing the explainability of the proposed user satisfaction prediction models to clarify why a user is dissatisfied can be an effective step towards the improvement of user experience in interacting with smart devices in physical spaces.

Our proposed Intent Sensitive Word Embeddings (ISWE) can be used as word representation for any natural language processing problem that has words as input. This presents new opportunities such as incorporating ISWE in language understanding problems such as slot tagging and intent classification. At last, incorporating user satisfaction as a metric to optimize task completion flows in IAs and improve their performance in a production environment remains a line of research for future work.

Apart from all the technical challenges, as we are modeling users interacting with smart devices, there are societal and ethical challenges too, in how to ensure privacy preserving personalization, fair personalization, explainable user models, and transparency in what data is shared or used locally. This is an important line of research, which is left for future work. Smart devices hold the promise to bring the powerful tools of the online world into the physical world, and our results highlight similarities and differences with user interactions in traditional search and recommendation settings, and help promote the user experience while interacting with smart devices.

Bibliography

[1] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer, 2011. (Cited on page 15.)

[2] M. Ageev, Q. Guo, D. Lagun, and E. Agichtein. Find it if you can: A game for modeling different types of web search success using interaction data. In *SIGIR*, pages 345–354, 2011. (Cited on pages 107 and 110.)

[3] Q. Ai, Y. Zhang, K. Bi, X. Chen, and W. B. Cro. Learning a hierarchical embedding model for personalized product search. In *SIGIR*, pages 645–654, 2017. (Cited on page 111.)

[4] M. Aliannejadi, I. Mele, and F. Crestani. Venue appropriateness prediction for contextual suggestion. In Voorhees and Ellis [161]. (Cited on pages 52 and 56.)

[5] J. Allan, B. Croft, A. Moffat, and M. Sanderson. Frontiers, challenges, and opportunities for information retrieval: Report from SWIRL 2012 the second strategic workshop on information retrieval in Lorne. *SIGIR Forum*, 46(1):2–32, 2012. (Cited on page 59.)

[6] S. Alletto, R. Cucchiara, G. D. Fiore, L. Mainetti, V. Mighali, L. Patrono, and G. Serra. An indoor location-aware system for an IoT-based smart museum. *IEEE Internet of Things Journal*, 3(2):244–253, 2016. (Cited on page 17.)

[7] M. ALMasri, C. Berrut, and J.-P. Chevallet. A comparison of deep learning based query expansion with pseudo-relevance feedback and mutual information. In *ECIR*, pages 709–715. Springer, 2016. (Cited on page 111.)

[8] H. Arasteh, V. Hosseinnezhad, V. Loia, A. Tommasetti, O. Troisi, M. Shafie-khah, and P. Siano. Iot-based smart cities: A survey. In *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, pages 1–6, 2016. (Cited on page 1.)

[9] L. Ardissono, T. Kuflik, and D. Petrelli. Personalization in cultural heritage: the road travelled and the one ahead. *User Modeling and User-Adapted Interaction*, 22(1):73–99, 2012. (Cited on page 16.)

[10] C. Ardito, P. Buono, M. F. Costabile, G. Desolda, R. Lanzilotti, M. Matera, and A. Piccinno. Towards enabling cultural-heritage experts to create customizable visit experiences. In *Proceedings of the 2018 AVI-CH Workshop on Advanced Visual Interfaces for Cultural Heritage*, volume 2091, 2018. (Cited on page 17.)

[11] K. Ashton. That internet of things thing. *RFID journal*, 22(7):97–114, 2009. (Cited on page 17.)

[12] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Computer networks*, 54(15): 2787–2805, 2010. (Cited on page 11.)

[13] E. Bakshy and D. Eckles. Uncertainty in online experiments with dependent data: An evaluation of bootstrap methods. In *KDD*, pages 1303–1311, 2013. (Cited on page 110.)

[14] P. Barnaghi, W. Wang, C. Henson, and K. Taylor. Semantics for the internet of things: early progress and back to the future. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 8 (1):1–21, 2012. (Cited on page 11.)

[15] I. Bartolini, V. Moscato, R. G. Pensa, A. Penta, A. Picariello, C. Sansone, and M. L. Sapino. Recommending multimedia visiting paths in cultural heritage applications. *Multimedia Tools and Applications*, 75(7):3813–3842, 2016. (Cited on page 16.)

[16] M. Bayomi and S. Lawless. ADAPT_TCD: An ontology-based context aware approach for contextual suggestion. In Voorhees and Ellis [161]. (Cited on page 53.)

[17] T. Benaglia, D. Chauveau, D. R. Hunter, and D. S. Young. mixtools: An R Package for Analyzing Finite Mixture Models. *Journal of Statistical Software*, 32(6):1–29, 2009. (Cited on pages 92 and 93.)

[18] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13 (1):281–305, 2012. (Cited on page 114.)

[19] B. Berjani and T. Strufe. A recommendation system for spots in location-based online social networks. In *Proceedings of the 4th Workshop on Social Network Systems*, SNS '11, pages 4:1–4:6. ACM, 2011. (Cited on page 15.)

[20] F. Bohnert, I. Zukerman, and J. Laures. GECKOmmender: Personalised theme and tour recommendations for museums. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 26–37. Springer, 2012. (Cited on page 16.)

[21] F. Brown, M. Lawrence, and V. Morrison. Conversational virtual healthcare assistant, Jan. 3 2017. URL `https://www.google.com/patents/US9536049`. US Patent 9,536,049. (Cited on pages 87 and 107.)

[22] C. Buckley, D. Dimmick, I. Soboroff, and E. Voorhees. Bias and the limits of pooling for large collections. *Information retrieval*, 10(6):491–508, 2007. (Cited on pages 50, 63, and 65.)

[23] J. Callan, J. Allan, C. L. A. Clarke, S. Dumais, D. A. Evans, M. Sanderson, and C. Zhai. Meeting of

the MINDS: An information retrieval research agenda. *SIGIR Forum*, 41(2):25–34, 2007. (Cited on page 59.)

[24] B. Carterette. On rank correlation and the distance between rankings. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 436–443, 2009. (Cited on page 65.)

[25] B. Carterette, J. Allan, and R. Sitaraman. Minimal test collections for retrieval evaluation. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 268–275. ACM, 2006. (Cited on page 62.)

[26] B. Carterette, E. Kanoulas, V. Pavlu, and H. Fang. Reusable test collections through experimental design. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 547–554, 2010. (Cited on page 62.)

[27] B. Carterette, A. Bah, and M. Zengin. Dynamic test collections for retrieval evaluation. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, ICTIR '15, pages 91–100. ACM, 2015. (Cited on page 63.)

[28] L. Catarinucci, D. de Donno, L. Mainetti, L. Palano, L. Patrono, M. L. Stefanizzi, and L. Tarricone. An IoT-aware architecture for smart healthcare systems. *IEEE Internet of Things Journal*, 2(6):515–526, 2015. (Cited on page 17.)

[29] L. D. Catledge and J. E. Pitkow. Characterizing browsing strategies in the world-wide web. *Computer Networks and ISDN Systems*, 27(6):1065 – 1073, 1995. Proceedings of the Third International World-Wide Web Conference. (Cited on pages 88, 90, and 91.)

[30] U. B. Ceipidor, C. M. Medaglia, V. Volpi, A. Moroni, S. Sposato, M. Carboni, and A. Caridi. NFC technology applied to touristic-cultural field: A case study on an italian museum. In *2013 5th International Workshop on Near Field Communication (NFC)*, pages 1–6, 2013. (Cited on page 17.)

[31] A. Chianese and F. Picciali. Designing a smart museum: When cultural heritage joins IoT. In *2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies*, pages 300–306, 2014. (Cited on page 17.)

[32] A. Chuklin, I. Markov, and M. d. Rijke. Click models for web search. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 7(3):1–115, 2015. (Cited on page 12.)

[33] C. W. Cleverdon. Report on the testing and analysis of an investigation into the comparative efficiency of indexing systems. Technical report, College of Aeronautics, Cranfield UK, 1962. (Cited on page 59.)

[34] R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1):5–32, 1999. (Cited on pages 88 and 89.)

[35] G. V. Cormack and T. R. Lynam. Power and bias of subset pooling strategies. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 837–838, 2007. (Cited on pages 65 and 77.)

[36] G. V. Cormack, C. R. Palmer, and C. L. A. Clarke. Efficient construction of large test collections. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 282–289, 1998. (Cited on page 62.)

[37] J. S. Culpepper, F. Diaz, and M. D. Smucker. Research frontiers in information retrieval: Report from the third strategic workshop on information retrieval in Lorne (SWIRL 2018). *SIGIR Forum*, 52(1), 2018. (Cited on page 59.)

[38] D. L. Davies and D. W. Bouldin. A cluster separation measure. *PAMI*, (2):224–227, 1979. (Cited on page 92.)

[39] A. Dean-Hall, C. L. A. Clarke, J. Kamps, P. Thomas, and E. M. Voorhees. Overview of the TREC 2012 contextual suggestion track. In E. M. Voorhees and L. P. Buckland, editors, *The Twenty-First Text REtrieval Conference Proceedings (TREC 2012)*. National Institute for Standards and Technology: NIST Special Publication 500-298, 2013. (Cited on page 41.)

[40] A. Dean-Hall, C. L. Clarke, J. Kamps, P. Thomas, and E. M. Voorhees. Overview of the TREC 2014 contextual suggestion track. In *Proceeding of Text REtrieval Conference (TREC)*, 2014. (Cited on page 60.)

[41] A. Dean-Hall, C. L. A. Clarke, J. Kamps, P. Thomas, and E. M. Voorhees. Overview of the TREC 2014 contextual suggestion track. In E. M. Voorhees and A. Ellis, editors, *Proceedings of the Twenty-Third Text REtrieval Conference (TREC 2014)*. National Institute for Standards and Technology, NIST Special Publication 500-308, 2015. (Cited on page 41.)

[42] A. Dean-Hall, C. L. A. Clarke, J. Kamps, J. Kiseleva, and E. M. Voorhees. Overview of the TREC 2015 contextual suggestion track. In E. M. Voorhees and A. Ellis, editors, *Proceedings of the Twenty-Fourth Text REtrieval Conference (TREC 2015)*. National Institute for Standards and Technology, NIST Special Publication 500-319, 2016. (Cited on pages 64 and 85.)

[43] A. Dean-Hall, C. L. A. Clarke, J. Kamps, J. Kiseleva, and E. M. Voorhees. Overview of the TREC 2015 contextual suggestion track. In E. M. Voorhees and A. Ellis, editors, *Proceedings of the Twenty-Fourth Text REtrieval Conference (TREC 2015)*. National Institute for Standards and Technology, NIST Special Publication 500-319, 2016. (Cited on page 41.)
[44] M. Dehghani, J. Kamps, H. Azarbonyad, and M. Marx. Significant words language models for contextual suggestion. In Voorhees and Ellis [161]. (Cited on page 54.)
[45] M. Dehghani, H. Zamani, A. Severyn, J. Kamps, and W. B. Croft. Neural ranking models with weak supervision. In *SIGIR*, pages 65–74, 2017. (Cited on page 111.)
[46] A. Deng, T. Li, and Y. Guo. Statistical inference in two-stage online controlled experiments with treatment selection and validation. In *WWW*, pages 609–618, 2014. (Cited on page 110.)
[47] F. Diaz, B. Mitra, and N. Craswell. Query expansion with locally-trained word embeddings. *ACL*, 2016. (Cited on pages 111 and 114.)
[48] A. Drutsa, G. Gusev, and P. Serdyukov. Future user engagement prediction and its application to improve the sensitivity of online experiments. In *WWW*, pages 256–266, 2015. (Cited on page 110.)
[49] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011. (Cited on page 26.)
[50] C. Eickhoff, P. Dekker, and A. P. de Vries. Supporting children's web search in school environments. In *IIIX*, IIIX '12, pages 129–137, 2012. (Cited on page 98.)
[51] C. Eickhoff, J. Teevan, R. White, and S. Dumais. Lessons from the journey: A query log analysis of within-session learning. In *WSDM*, pages 223–232, 2014. (Cited on pages 27, 88, 90, 91, and 98.)
[52] O. Evangelatos, K. Samarasinghe, and J. Rolim. Syndesi: A framework for creating personalized smart environments using wireless sensor networks. In *2013 IEEE International Conference on Distributed Computing in Sensor Systems*, pages 325–330, 2013. (Cited on page 17.)
[53] M. Fadaee, A. Bisazza, and C. Monz. Learning topic-sensitive word representations. *ACL*, 2017. (Cited on pages 111, 114, and 116.)
[54] H. A. Feild, J. Allan, and R. Jones. Predicting searcher frustration. In *SIGIR*, pages 34–41, 2010. (Cited on pages 107, 110, and 118.)
[55] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Trans. Inf. Syst.*, 23(2):147–168, 2005. (Cited on pages 108 and 110.)
[56] A. Freeman, S. Adams Becker, M. Cummins, E. McKelroy, C. Giesinger, and B. Yuhnke. *The NMC horizon report: 2016: Museum edition*. The New Media Consortium, Austin, Texas, 2016. (Cited on page 38.)
[57] P. Friess. *Internet of things: converging technologies for smart environments and integrated ecosystems*. River Publishers, 2013. (Cited on page 11.)
[58] D. Ganguly, D. Roy, M. Mitra, and G. J. Jones. Word embedding based generalized language model for information retrieval. In *SIGIR*, pages 795–798, 2015. (Cited on page 111.)
[59] H. Gao, J. Tang, X. Hu, and H. Liu. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 93–100. ACM, 2013. (Cited on page 15.)
[60] A. Graves et al. *Supervised sequence labelling with recurrent neural networks*, volume 385. Springer, 2012. (Cited on page 113.)
[61] M. Gribaudo, M. Iacono, and A. H. Levis. An IoT-based monitoring approach for cultural heritage sites: The Matera case. *Concurrency and Computation: Practice and Experience*, 29(11), 2017. (Cited on page 17.)
[62] K. Grieser, T. Baldwin, and S. Bird. Dynamic path prediction and recommendation in a museum environment. In *Prococeedings of the Workshop on Language Technology for Cultural Heritage Data (LaT-eCH 2007)*, pages 49–56, 2007. (Cited on page 16.)
[63] J.-B. Griesner, T. Abdessalem, and H. Naacke. POI recommendation: Towards fused matrix factorization with geographical and temporal influences. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, pages 301–304. ACM, 2015. (Cited on page 15.)
[64] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645 – 1660, 2013. (Cited on page 17.)
[65] I. Guy. The role of user location in personalized search and recommendation. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, pages 236–236. ACM, 2015. (Cited on page 15.)
[66] A. Halfaker, O. Keyes, D. Kluver, J. Thebault-Spieker, T. Nguyen, K. Shores, A. Uduwage, and M. Warncke-Wang. User session identification based on strong regularities in inter-activity time. In

WWW, WWW '15, pages 410–418, 2015. (Cited on pages 88, 89, 90, 91, 92, 93, and 94.)

[67] S. H. Hashemi and J. Kamps. Venue recommendation and web search based on anchor text. In *Proceeding of Text REtrieval Conference (TREC)*, 2014. (Cited on pages 9, 48, and 71.)

[68] S. H. Hashemi and J. Kamps. Where to go next?: Exploiting behavioral user models in smart environments. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 50–58. ACM, 2017. (Cited on pages vi, 7, 14, 17, 29, and 107.)

[69] S. H. Hashemi and J. Kamps. On the reusability of personalized test collections. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 185–189, 2017. (Cited on page 9.)

[70] S. H. Hashemi and J. Kamps. Skip or stay: Users behavior in dealing with onsite information interaction crowd-bias. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, pages 389–392, 2017. (Cited on pages vi, 7, 11, 13, 16, 22, 29, and 33.)

[71] S. H. Hashemi and J. Kamps. Exploiting behavioral user models for point of interest recommendation in smart museums. *New Review of Hypermedia and Multimedia*, 24(3):228–261, 2018. (Cited on pages vi, 7, and 134.)

[72] S. H. Hashemi, C. L. Clarke, A. Dean-Hall, J. Kamps, and J. Kiseleva. On the reusability of open test collections. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 827–830, 2015. (Cited on pages vi, 8, 42, 48, 49, 56, 61, 64, 66, 67, 68, 69, and 129.)

[73] S. H. Hashemi, M. Dehghani, and J. Kamps. Parsimonious user and group profiling in venue recommendation. In *Proceeding of Text REtrieval Conference (TREC)*, 2015. (Cited on page 9.)

[74] S. H. Hashemi, C. L. Clarke, A. Dean-Hall, J. Kamps, and J. Kiseleva. An easter egg hunting approach to test collection building in dynamic domains. In *NTCIR-EVIA*, pages 1–8, 2016. (Cited on pages vi, 8, 42, 48, 56, 61, and 129.)

[75] S. H. Hashemi, C. L. A. Clarke, A. Dean-Hall, J. Kamps, and J. Kiseleva. Test collection building and maintenance in dynamic domains. In *15th Dutch-Belgian Information Retrieval Workshop (DIR)*, 2016. (Cited on pages 48 and 64.)

[76] S. H. Hashemi, C. L. A. Clarke, J. Kamps, J. Kiseleva, and E. M. Voorhees. Overview of the TREC 2016 contextual suggestion track. In *Proceeding of Text REtrieval Conference (TREC)*, 2016. (Cited on pages vi, 7, 15, 64, 85, 87, and 107.)

[77] S. H. Hashemi, W. Hupperetz, J. Kamps, and M. van der Vaart. Effects of position and time bias on understanding onsite users' behavior. In *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval*, CHIIR '16, pages 277–280. ACM, 2016. (Cited on pages vi, 7, 11, 13, 22, 24, 29, and 33.)

[78] S. H. Hashemi, J. Kamps, and N. O. Amer. Neural endorsement based contextual suggestion. In *Proceeding of Text REtrieval Conference (TREC)*, 2016. (Cited on pages 9 and 56.)

[79] S. H. Hashemi, J. Kamps, and W. Hupperetz. Busy versus empty museums: Effects of visitors crowd on users behaviors in smart museums. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 333–334, 2017. (Cited on page 9.)

[80] S. H. Hashemi, K. Williams, A. El Kholy, I. Zitouni, and P. Crook. Impact of domain and user's learning phase on task and session identification in smart speaker intelligent assistants. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1193–1202, 2018. (Cited on pages vii, 8, 12, 112, 117, and 122.)

[81] S. H. Hashemi, K. Williams, A. El Kholy, I. Zitouni, and P. Crook. Measuring user satisfaction on smart speaker intelligent assistants using intent sensitive query embeddings. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1183–1192, 2018. (Cited on pages vii, 1, 8, and 12.)

[82] S. H. Hashemi, C. L. Clarke, A. Dean-Hall, J. Kamps, and J. Kiseleva. An analysis of test collection building in dynamic domains. *Under Submission*, 2020. (Cited on pages vi and 8.)

[83] A. Hassan. A semi-supervised approach to modeling web search satisfaction. In *SIGIR*, pages 275–284, 2012. (Cited on pages 107, 110, 111, and 119.)

[84] A. Hassan and R. W. White. Personalized models of search satisfaction. In *CIKM*, pages 2009–2018, 2013. (Cited on pages 98, 103, 108, and 110.)

[85] A. Hassan, R. Jones, and K. L. Klinkner. Beyond dcg: User behavior as a predictor of a successful search. In *WSDM*, pages 221–230, 2010. (Cited on pages 107, 110, and 111.)

[86] A. Hassan, X. Shi, N. Craswell, and B. Ramsey. Beyond clicks: query reformulation as a predictor of search satisfaction. In *CIKM*, pages 2019–2028, 2013. (Cited on pages 108, 110, 111, and 119.)

[87] D. He, A. Göker, and D. J. Harper. Combining evidence for automatic web session identification. *Inf.*

Process. Manage., 38(5):727742, 2002. (Cited on pages 89 and 90.)
[88] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53, 2004. (Cited on page 29.)
[89] J. M. Hernández-Muñoz, J. B. Vercher, L. Muñoz, J. A. Galache, M. Presser, L. A. H. Gómez, and J. Pettersson. Smart cities at the forefront of the future internet. In *The Future Internet Assembly*, pages 447–462. Springer, 2011. (Cited on page 11.)
[90] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012. (Cited on pages 26 and 114.)
[91] H. C. Huurdeman, J. Kamps, T. Samar, A. P. de Vries, A. Ben-David, and R. A. Rogers. Lost but not forgotten: finding pages on the unarchived web. *International Journal on Digital Libraries*, 16(3): 247–265, 2015. (Cited on page 59.)
[92] B. J. Jansen, A. Spink, C. Blakely, and S. Koshman. Defining a session on web search engines: Research articles. *JASIST*, 58(6):862–871, 2007. (Cited on pages 88, 89, and 91.)
[93] G. K. Jayasinghe, W. Webber, M. Sanderson, and J. S. Culpepper. Improving test collection pools with machine learning. In *Proceedings of the 2014 Australasian Document Computing Symposium*, 2014. (Cited on page 62.)
[94] J. Jiang, A. Hassan Awadallah, R. Jones, U. Ozertem, I. Zitouni, R. Gurunath Kulkarni, and O. Z. Khan. Automatic online evaluation of intelligent assistants. In *WWW*, pages 506–516, 2015. (Cited on pages 87 and 108.)
[95] J. Jiang, A. Hassan Awadallah, X. Shi, and R. W. White. Understanding and predicting graded search satisfaction. In *WSDM*, pages 57–66, 2015. (Cited on pages 107, 108, and 110.)
[96] L. Johnson, S. Adams Becker, V. Estrada, and A. Freeman. *The NMC Horizon Report: 2015 Museum Edition.* The New Media Consortium, Austin, Texas, 2015. (Cited on page 38.)
[97] R. Jones and K. L. Klinkner. Beyond the session timeout: Automatic hierarchical segmentation of search topics in query logs. In *CIKM*, CIKM '08, pages 699–708. ACM, 2008. (Cited on pages 88, 90, and 91.)
[98] G. Kalamatianos and A. Arampatzis. Recommending points-of-interest via weighted kNN, rated rocchio, and borda count fusion. In Voorhees and Ellis [161]. (Cited on page 54.)
[99] J. Kamps, S. Geva, C. Peters, T. Sakai, A. Trotman, and E. Voorhees. Report on the sigir 2009 workshop on the future of IR evaluation. *SIGIR Forum*, 43(2):13–23, 2009. (Cited on page 59.)
[100] M. Kamvar, M. Kellar, R. Patel, and Y. Xu. Computers and iphones and mobile phones, oh my!: A logs-based comparison of search users on different devices. In *WWW*, pages 801–810, 2009. (Cited on pages 87 and 107.)
[101] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 79–86. ACM, 2010. (Cited on page 15.)
[102] D. Kelly. Methods for evaluating interactive information retrieval systems with users. *Foundations and Trends in Information Retrieval*, 3(1-2):1–224, 2009. (Cited on pages 108 and 110.)
[103] M. Khabsa, A. Crook, A. H. Awadallah, I. Zitouni, T. Anastasakos, and K. Williams. Learning to account for good abandonment in search success metrics. In *CIKM*, pages 1893–1896, 2016. (Cited on pages 87 and 108.)
[104] M. Khorasani, H. Sadjadi, F. Ramazani, and F. Ensan. A context based recommender system through collaborative filtering and word embedding techniques. In Voorhees and Ellis [161]. (Cited on page 54.)
[105] Y. Kim, A. Hassan, R. W. White, and Y.-M. Wang. Playing by the rules: Mining query associations to predict search performance. In *WSDM*, pages 133–142, 2013. (Cited on pages 107 and 110.)
[106] Y. Kim, A. Hassan, R. W. White, and I. Zitouni. Comparing client and server dwell time estimates for click-level satisfaction prediction. In *SIGIR*, pages 895–898, 2014. (Cited on pages 108 and 110.)
[107] Y. Kim, A. Hassan, R. W. White, and I. Zitouni. Modeling dwell time to predict click-level satisfaction. In *WSDM*, pages 193–202, 2014. (Cited on pages 107, 108, and 110.)
[108] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. (Cited on page 114.)
[109] J. Kiseleva, K. Williams, A. Hassan Awadallah, A. C. Crook, I. Zitouni, and T. Anastasakos. Predicting user satisfaction with intelligent assistants. In *SIGIR*, pages 45–54. ACM, 2016. (Cited on pages 87, 91, 107, 108, 110, and 112.)
[110] J. Kiseleva, K. Williams, J. Jiang, A. Hassan Awadallah, A. C. Crook, I. Zitouni, and T. Anastasakos.

Understanding user satisfaction with intelligent assistants. In *CHIIR*, pages 121–130. ACM, 2016. (Cited on pages 87, 107, 108, and 110.)

[111] D. Lagun, C.-H. Hsieh, D. Webster, and V. Navalpakkam. Towards better measurement of attention and satisfaction in mobile search. In *SIGIR*, pages 113–122, 2014. (Cited on pages 108 and 110.)

[112] J. Lanir, T. Kuflik, E. Dim, A. J. Wecker, and O. Stock. The influence of a location-aware mobile guide on museum visitors' behavior. *Interacting with Computers*, 25(6):443–460, 2013. (Cited on pages 16 and 37.)

[113] A. W. Lazonder, H. J. Biemans, and I. G. Wopereis. Differences between novice and experienced users in searching information on the world wide web. *Journal of the American Society for Information Science*, 51(6):576–581, 2000. (Cited on page 98.)

[114] O. Levy and Y. Goldberg. Dependency-based word embeddings. In *ACL*, pages 302–308, 2014. (Cited on pages 111, 115, and 116.)

[115] X. Liu and K. Aberer. SoCo: A social network aided context-aware recommender system. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, pages 781–802. ACM, 2013. (Cited on page 15.)

[116] C. Lucchese, R. Perego, F. Silvestri, H. Vahabi, and R. Venturini. How random walks can help tourism. In *Proceedings of Advances in Information Retrieval: 34th European Conference on IR Research, ECIR 2012*, 2012. (Cited on pages 29 and 33.)

[117] R. Mehrotra and E. Yilmaz. Task embeddings: Learning query embeddings using task context. In *CIKM*, CIKM '17, pages 2199–2202, 2017. (Cited on page 111.)

[118] R. Mehrotra, A. E. Kholy, I. Zitouni, M. Shokouhi, and A. Hassan. Identifying user sessions in interactions with intelligent digital assistants. In *WWW*, WWW '17 Companion, pages 821–822, 2017. (Cited on pages 88, 90, 91, 93, 94, 96, and 100.)

[119] R. Mehrotra, I. Zitouni, A. H. Awadallah, A. El Kholy, and M. Khabsa. User interaction sequences for search satisfaction prediction. In *SIGIR*, pages 165–174. ACM, 2017. (Cited on pages 87, 88, 95, 107, 108, 110, 111, and 118.)

[120] D. Mehrzadi and D. G. Feitelson. On extracting session data from activity logs. In *SYSTOR*, SYSTOR '12, pages 3:1–3:7. ACM, 2012. (Cited on page 90.)

[121] V. Mighali, G. Del Fiore, L. Patrono, L. Mainetti, S. Alletto, G. Serra, and R. Cucchiara. Innovative IoT-aware services for a smart museum. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 547–550. ACM, 2015. (Cited on page 17.)

[122] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119. 2013. (Cited on pages 111, 115, and 116.)

[123] T. Minka et al. Divergence measures and message passing. Technical report, Technical report, Microsoft Research, 2005. (Cited on page 97.)

[124] J. Mo, L. Lamontagne, and R. Khoury. Word embeddings and global preference for contextual suggestion. In Voorhees and Ellis [161]. (Cited on page 55.)

[125] A. Moffat, W. Webber, and J. Zobel. Strategic system comparisons via targeted relevance judgments. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 375–382, 2007. (Cited on page 62.)

[126] A. L. Montgomery and C. Faloutsos. Identifying web browsing trends and patterns. *Computer*, 34(7): 94–95, 2001. (Cited on page 90.)

[127] P. Mtshali and F. Khubisa. A smart home appliance control system for physically disabled people. In *2019 Conference on Information Communications Technology and Society (ICTAS)*, pages 1–5, 2019. (Cited on page 1.)

[128] J. Mueller and A. Thyagarajan. Siamese recurrent architectures for learning sentence similarity. In *AAAI*, pages 2786–2792, 2016. (Cited on page 113.)

[129] M. Nadjarbashi-Noghani and A. A. Ghorbani. Improving the referrer-based web log session reconstruction. In *Communication Networks and Services Research*, pages 286–292. IEEE, 2004. (Cited on page 89.)

[130] L. Niu, X. Dai, J. Zhang, and J. Chen. Topic2vec: learning distributed representations of topics. In *IALP*, pages 193–196. IEEE, 2015. (Cited on page 116.)

[131] M. P. O'Mahony, N. J. Hurley, and G. C. M. Silvestre. Recommender systems: Attack types and strategies. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 1*, AAAI'05, pages 334–339, 2005. (Cited on page 84.)

[132] H. C. Ozmutlu and F. avdur. Application of automatic topic identification on excite web search engine data logs. *IPM*, 41(5):1243 – 1262, 2005. (Cited on pages 89 and 90.)

[133] S. Ozmutlu. Automatic new topic identification using multiple linear regression. *IPM*, 42(4):934 – 950, 2006. (Cited on pages 89 and 90.)
[134] M.-H. Park, J.-H. Hong, and S.-B. Cho. Location-based recommendation system using bayesian user's preference model in mobile devices. In *International Conference on Ubiquitous Intelligence and Computing*, pages 1130–1139. Springer, 2007. (Cited on page 15.)
[135] A. Passos, V. Kumar, and A. McCallum. Lexicon infused phrase embeddings for named entity resolution. *ACL*, 2014. (Cited on page 111.)
[136] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys & Tutorials*, 16(1):414–454, 2014. (Cited on page 11.)
[137] K. Radinsky, K. M. Svore, S. T. Dumais, M. Shokouhi, J. Teevan, A. Bocharov, and E. Horvitz. Behavioral dynamics on the web: Learning, modeling, and prediction. *ACM TOIS*, 31(3):16:1–16:37, 2013. (Cited on page 84.)
[138] F. Radlinski and T. Joachims. Query chains: Learning to rank from implicit feedback. In *KDD*, KDD '05, pages 239–248. ACM, 2005. (Cited on pages 89 and 90.)
[139] F. Radlinski, M. Szummer, and N. Craswell. Inferring query intent from reformulations and clicks. In *WWW*, pages 1171–1172, 2010. (Cited on page 109.)
[140] A. S. Rao, A. V. Sharma, and C. S. Narayan. A context aware system for an IoT-based smart museum. In *2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*, pages 1–5, 2017. (Cited on page 17.)
[141] N. Rekabsaz, M. Lupu, A. Hanbury, and H. Zamani. Word embedding causes topic shifting; exploit global context! In *SIGIR*, pages 1105–1108, 2017. (Cited on page 111.)
[142] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 635–644. ACM, 2011. (Cited on page 15.)
[143] T. Sakai. Comparing metrics across TREC and NTCIR: The robustness to system bias. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 581–590, 2008. (Cited on page 63.)
[144] C. Scholz, J. Illig, M. Atzmueller, and G. Stumme. On the predictability of talk attendance at academic conferences. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media*, pages 279–284. ACM, 2014. (Cited on page 15.)
[145] M. Shokouhi, U. Ozertem, and N. Craswell. Did you say U2 or YouTube?: Inferring implicit transcripts from voice search logs. In *WWW*, pages 1215–1224, 2016. (Cited on pages 27, 88, 90, and 91.)
[146] M. Silverio-Fernández, S. Renukappa, and S. Suresh. What is a smart device? - a conceptualisation within the paradigm of the internet of things. *Visualization in Engineering*, 6(1), 2018. (Cited on page 1.)
[147] I. Soboroff. Dynamic test collections: Measuring search effectiveness on the live web. In *SIGIR*, pages 276–283, 2006. (Cited on pages 59, 63, and 70.)
[148] I. Soboroff and S. Robertson. Building a filtering test collection for trec 2002. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 243–250, 2003. (Cited on page 62.)
[149] I. Soboroff, C. Nicholas, and P. Cahan. Ranking retrieval systems without relevance judgments. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 66–73. ACM, 2001. (Cited on page 62.)
[150] K. Sornalatha and V. R. Kavitha. IoT based smart museum using bluetooth low energy. In *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, pages 520–523, 2017. (Cited on page 17.)
[151] K. Sparck Jones and C. J. Van Rijsbergen. Report on the need for and provision of an 'ideal' information retrieval test collection. Technical report, University Computer Laboratory, Cambridge, 1975. (Cited on page 62.)
[152] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014. (Cited on pages 26 and 114.)
[153] M. Strohbach, H. Ziekow, V. Gazis, and N. Akiva. *Towards a Big Data Analytics Framework for IoT and Smart City Applications*, pages 257–282. 2015. (Cited on page 17.)
[154] D. Tang, A. Agarwal, D. O'Brien, and M. Meyer. Overlapping experiment infrastructure: More, better, faster experimentation. In *KDD*, pages 17–26, 2010. (Cited on page 110.)

[155] J. Teevan, A. Karlson, S. Amini, A. J. B. Brush, and J. Krumm. Understanding the importance of location, time, and people in mobile local search behavior. In *MobileHCI*, pages 77–80, 2011. (Cited on pages 87 and 107.)

[156] TREC. Contextual suggestion track. `https://sites.google.com/site/treccontext/`. (Cited on page 60.)

[157] P. Vakkari, M. Pennanen, and S. Serola. Changes of search terms and tactics while writing a research proposal: A longitudinal case study. *IPM*, 39(3):445 – 463, 2003. (Cited on page 98.)

[158] M. van Zeelt, F. den Hengst, and S. H. Hashemi. Collecting high-quality dialogue user satisfaction ratings with third-party annotators. In *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval*, pages 363–367, 2020. (Cited on page 9.)

[159] E. M. Voorhees. Evaluation by highly relevant documents. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 74–82, 2001. (Cited on pages 66 and 77.)

[160] E. M. Voorhees. The philosophy of information retrieval evaluation. In *CLEF*, pages 355–370, 2002. (Cited on page 63.)

[161] E. M. Voorhees and A. Ellis, editors. *Proceedings of the Twenty-Fifth Text REtrieval Conference (TREC 2016)*, 2017. National Institute for Standards and Technology, NIST Special Publication 500-321. (Cited on pages 137, 139, 141, 142, and 144.)

[162] E. M. Voorhees, D. K. Harman, et al. Experiment and evaluation in information retrieval. *MIT press*, 2005. (Cited on page 59.)

[163] G. Wang, X. Zhang, S. Tang, H. Zheng, and B. Y. Zhao. Unsupervised clickstream clustering for user behavior analysis. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 225–236. ACM, 2016. (Cited on page 12.)

[164] A. J. Wecker, J. Lanir, T. Kuflik, and O. Stock. Pathlight: Supporting navigation of small groups in the museum context. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 569–574. ACM, 2011. (Cited on page 16.)

[165] R. W. White, S. T. Dumais, and J. Teevan. Characterizing the influence of domain expertise on web search behavior. In *WSDM*, WSDM '09, pages 132–141. ACM, 2009. (Cited on page 98.)

[166] B. M. Wildemuth. The effects of domain knowledge on search tactic formulation. *JASIST*, 55(3): 246–258, 2004. (Cited on page 98.)

[167] K. Williams and I. Zitouni. Does that mean you're happy? rnn-based modeling of user interaction sequences to detect good abandonment. In *CIKM*. ACM, 2017. (Cited on pages 113 and 117.)

[168] K. Williams, J. Kiseleva, A. C. Crook, I. Zitouni, A. H. Awadallah, and M. Khabsa. Is this your final answer?: Evaluating the effect of answers on good abandonment in mobile search. In *SIGIR*, pages 889–892. ACM, 2016. (Cited on pages 87, 108, and 110.)

[169] K. Williams, J. Kiseleva, A. C. Crook, I. Zitouni, A. H. Awadallah, and M. Khabsa. Detecting good abandonment in mobile search. In *WWW*, pages 495–505, 2016. (Cited on pages 87, 107, 108, 110, and 118.)

[170] K. Williams, S. H. Hashemi, and I. Zitouni. Automatic task completion flows from web APIs. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1009–1012, 2019. (Cited on pages 9 and 134.)

[171] X. Xiao, Y. Zheng, Q. Luo, and X. Xie. Finding similar users using category-based location history. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, pages 442–445. ACM, 2010. (Cited on page 15.)

[172] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 325–334. ACM, 2011. (Cited on page 15.)

[173] E. Yilmaz, J. A. Aslam, and S. Robertson. A new rank correlation coefficient for information retrieval. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 587–594, 2008. (Cited on page 65.)

[174] D. Yin, S. Gao, Z. Peng, Y. Li, and R. Liu. Beijing university of posts and telecommunications (BUPT) at TREC 2016: A rating model based on tags for contextual suggestion. In Voorhees and Ellis [161]. (Cited on page 56.)

[175] J. J.-C. Ying, E. H.-C. Lu, W.-N. Kuo, and V. S. Tseng. Urban point-of-interest recommendation by mining user check-in behaviors. In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, UrbComp '12, pages 63–70. ACM, 2012. (Cited on page 15.)

[176] H. Zamani and W. B. Croft. Estimating embedding vectors for queries. In *ICTIR*, pages 123–132,

2016. (Cited on page 111.)
[177] H. Zamani and W. B. Croft. Embedding-based query language models. In *ICTIR*, pages 147–156, 2016. (Cited on page 111.)
[178] H. Zamani and W. B. Croft. Relevance-based word embedding. In *SIGIR*, pages 505–514, 2017. (Cited on page 111.)
[179] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of things for smart cities. *IEEE Internet of Things Journal*, 1(1):22–32, 2014. (Cited on page 17.)
[180] G. Zheng and J. Callan. Learning to reweight terms with distributed representations. In *SIGIR*, pages 575–584, 2015. (Cited on page 111.)
[181] V. W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang. Collaborative filtering meets mobile recommendation: A user-centered approach. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, pages 236–241. AAAI Press, 2010. (Cited on page 15.)
[182] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with GPS history data. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 1029–1038. ACM, 2010. (Cited on page 15.)
[183] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 791–800. ACM, 2009. (Cited on page 15.)
[184] G. Zhou, T. He, J. Zhao, and P. Hu. Learning continuous word embedding with metadata for question retrieval in community question answering. In *ACL (1)*, pages 250–259, 2015. (Cited on page 111.)
[185] J. Zhuang, T. Mei, S. C. Hoi, Y.-Q. Xu, and S. Li. When recommendation meets mobile: Contextual and personalized recommendation on the go. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, UbiComp '11, pages 153–162. ACM, 2011. (Cited on page 15.)
[186] J. Zobel. How reliable are the results of large-scale information retrieval experiments? In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 307–314, 1998. (Cited on pages 62, 63, and 65.)
[187] O. Zoeter. Recommendations in travel. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, pages 234–234. ACM, 2015. (Cited on page 15.)

Summary

Modeling users interacting with smart devices

Personalizing users' experience and the ability to perform complex tasks on smart devices and environments such as smart speakers and smart homes are changing the way people are doing their daily tasks. Checking the weather and planning to visit a museum is as simple as asking your smart speaker at home to read out loud the weather condition and commanding the Intelligent Assistant (IA) integrated with the smart speaker to book a ticket to visit the museum. To improve user experience in physical spaces such as smart homes, museums, and cities while performing their daily tasks, effective modeling of users interacting with smart devices is required.

The overall goal of this thesis is to improve users' experience in physical spaces such as smart cities and environments by modeling user interactions with smart devices. In Chapter 2, we model users' behavior in interacting with smart devices in a smart museum to recommend unseen archaeological objects to visit without asking users to provide any information about their preferences. To understand users' preferences, we have studied both users' onsite physical interaction behavior in the physical space and their online digital interaction behavior at the search engine of the museum. We found similarities and differences in users' online and onsite interaction behaviors, which leads to incorporating both online digital and onsite physical user interactions in training an effective point of interest recommender system for a smart museum.

In Chapter 3 and Chapter 4 of the thesis, we focus on creating and maintaining reusable test collections for the evaluation of contextual suggestion systems to rank tourist attractions for users in a smart city context. Creating and maintaining a reusable test collection for offline evaluation of personalized contextual suggestion systems is challenging due to the personalized and dynamic nature of the test collection. However, personalization is an important aspect of contextual suggestion systems as it can have a direct impact on the user experience. Thus, we create a reusable test collection for the evaluation of personalized contextual suggestion systems and proposed an approach for maintaining the reusability of dynamic test collections.

Furthermore, to measure how satisfied users are in using smart devices such as smart speakers in their smart homes, Chapter 5 of this thesis is allocated to identifying tasks and sessions on smart speaker IAs using a time-oriented approach in analyzing smart speaker IA interaction logs. Then, Chapter 6 details our proposed user satisfaction prediction model on smart speaker IAs to measure user satisfaction while performing a task to improve users' experience in using smart speakers. In this thesis, we show how different contextual factors such as the learning curve of users have impacts on users' behavior on smart speaker IAs that lead to different task and session boundaries estimation on different contextual situations. Furthermore, we propose users' utterance intent as a signal to measure user satisfaction on smart speaker IAs and show how incorporating users' intent in query representation learning can improve a user satisfaction prediction model and consequently users' experience with smart speakers.

Samenvatting

Het modelleren van gebruikers en hun interacties met slimme apparaten

Slimme apparaten en omgevingen, zoals slimme speakers en slimme huizen, veranderen dagelijkse taken als het uitvragen van het weerbericht en het plannen van een museumbezoek. Een weerbericht kan bij een slimme speaker worden opgevraagd en een museumticket kan via een Intelligente Assistent (IA) op zon speaker worden gekocht. Zulke apparaten kunnen complexe taken uitvoeren en bieden een gepersonaliseerde gebruikerservaring. Om de dagelijkse gebruikerservaring van slimme apparaten in fysieke ruimtes zoals huizen, musea en steden te verbeteren is het nodig om gebruikers en hun interactie met slimme apparaten te modelleren.

The primaire doel van dit proefschrift is om de gebruikerservaring in fysieke ruimtes zoals slimme steden en slimme omgevingen te verbeteren door het modelleren van gebruikersinteracties met slimme apparaten. Hoofdstuk twee beschrijft een gedragsmodel van gebruikers van slimme apparaten in een slim museum. Het wordt gebruikt om archeologische voorwerpen die een gebruiker nog niet heeft gezien aan te bevelen. Hierbij hoeft de gebruiker zelf haar voorkeuren niet op te geven. Om deze voorkeuren toch te begrijpen, hebben we zowel fysieke interacties in het museum als digitale interacties met de zoekmachine van het museum bestudeerd. We vonden overeenkomsten en verschillen tussen het fysieke en digitale gedrag en gebruikten daarom beide bij het trainen van een aanbevelingssysteem voor interessepunten in het museum.

Hoofdstukken drie en vier gaan over het samenstellen en onderhouden van herbruikbare test collecties voor het evalueren van contextuele suggestiesystemen die toeristische attracties in een slimme stad voor een gebruiker rangschikken. De collecties kunnen gebruikt worden om contextuele aanbevelingssystemen offline te evalueren. Het maken en onderhouden van zulke collecties is ingewikkeld doordat ze gepersonaliseerd en dynamisch zijn. Het gepersonaliseerde karakter kan een grote invloed hebben op de gebruikerservaring van contextuele aanbevelingssystemen. Daarom presenteren we een herbruikbare test collectie voor het evalueren van gepersonaliseerde contextuele aanbevelingssystemen en stellen we een aanpak voor het behouden van de herbruikbaarheid van dynamische test collecties voor.

Hoofdstuk vijf is toegewijd aan het meten van de tevredenheid van gebruikers van slimme speakers in slimme huizen. Specifiek worden uit interactielogs van gebruikers met IAs taak- en sessiegrenzen herkend met een tijd-gebaseerde aanpak. Hierna volgt in hoofdstuk zes een model voor het voorspellen van gebruikerstevredenheid tijdens het uitvoeren van een taak met een slimme speaker. Doel is om de gebruikerservaring te verbeteren. We laten in dit proefschrift zien hoe verschillende contextuele factoren zoals de leercurve van gebruikers hun interactie met een IAs beïnvloeden en hoe deze factoren zich verhouden tot taak- en sessiegrenzen in verschillende contextuele situaties. Verder stellen we voor om uitingen van gebruikersintenties als signaal voor gebruikerstevredenheid in te zetten en laten we zien hoe het meenemen van gebruikersintenties bij het leren van een voorstelling van de gebruikersvraag kan helpen bij het voorspellen van de gebruikerstevredenheid en zo dus ook de gebruikerservaring met slimme speaker

verbeterd.

Titles in the ILLC Dissertation Series:

ILLC DS-2016-01: **Ivano A. Ciardelli**
Questions in Logic

ILLC DS-2016-02: **Zoé Christoff**
Dynamic Logics of Networks: Information Flow and the Spread of Opinion

ILLC DS-2016-03: **Fleur Leonie Bouwer**
What do we need to hear a beat? The influence of attention, musical abilities, and accents on the perception of metrical rhythm

ILLC DS-2016-04: **Johannes Marti**
Interpreting Linguistic Behavior with Possible World Models

ILLC DS-2016-05: **Phong Lê**
Learning Vector Representations for Sentences - The Recursive Deep Learning Approach

ILLC DS-2016-06: **Gideon Maillette de Buy Wenniger**
Aligning the Foundations of Hierarchical Statistical Machine Translation

ILLC DS-2016-07: **Andreas van Cranenburgh**
Rich Statistical Parsing and Literary Language

ILLC DS-2016-08: **Florian Speelman**
Position-based Quantum Cryptography and Catalytic Computation

ILLC DS-2016-09: **Teresa Piovesan**
Quantum entanglement: insights via graph parameters and conic optimization

ILLC DS-2016-10: **Paula Henk**
Nonstandard Provability for Peano Arithmetic. A Modal Perspective

ILLC DS-2017-01: **Paolo Galeazzi**
Play Without Regret

ILLC DS-2017-02: **Riccardo Pinosio**
The Logic of Kant's Temporal Continuum

ILLC DS-2017-03: **Matthijs Westera**
Exhaustivity and intonation: a unified theory

ILLC DS-2017-04: **Giovanni Cinà**
Categories for the working modal logician

ILLC DS-2017-05: **Shane Noah Steinert-Threlkeld**
Communication and Computation: New Questions About Compositionality

ILLC DS-2017-06: **Peter Hawke**
The Problem of Epistemic Relevance

ILLC DS-2017-07: **Aybüke Özgün**
Evidence in Epistemic Logic: A Topological Perspective

ILLC DS-2017-08: **Raquel Garrido Alhama**
Computational Modelling of Artificial Language Learning: Retention, Recognition & Recurrence

ILLC DS-2017-09: **Miloš Stanojević**
Permutation Forests for Modeling Word Order in Machine Translation

ILLC DS-2018-01: **Berit Janssen**
Retained or Lost in Transmission? Analyzing and Predicting Stability in Dutch Folk Songs

ILLC DS-2018-02: **Hugo Huurdeman**
Supporting the Complex Dynamics of the Information Seeking Process

ILLC DS-2018-03: **Corina Koolen**
Reading beyond the female: The relationship between perception of author gender and literary quality

ILLC DS-2018-04: **Jelle Bruineberg**
Anticipating Affordances: Intentionality in self-organizing brain-body-environment systems

ILLC DS-2018-05: **Joachim Daiber**
Typologically Robust Statistical Machine Translation: Understanding and Exploiting Differences and Similarities Between Languages in Machine Translation

ILLC DS-2018-06: **Thomas Brochhagen**
Signaling under Uncertainty

ILLC DS-2018-07: **Julian Schlöder**
Assertion and Rejection

ILLC DS-2018-08: **Srinivasan Arunachalam**
Quantum Algorithms and Learning Theory

ILLC DS-2018-09: **Hugo de Holanda Cunha Nobrega**
Games for functions: Baire classes, Weihrauch degrees, transfinite computations, and ranks

ILLC DS-2018-10: **Chenwei Shi**
Reason to Believe

ILLC DS-2018-11: **Malvin Gattinger**
New Directions in Model Checking Dynamic Epistemic Logic

ILLC DS-2018-12: **Julia Ilin**
Filtration Revisited: Lattices of Stable Non-Classical Logics

ILLC DS-2018-13: **Jeroen Zuiddam**
Algebraic complexity, asymptotic spectra and entanglement polytopes

ILLC DS-2019-01: **Carlos Vaquero**
What Makes A Performer Unique? Idiosyncrasies and commonalities in expressive music performance

ILLC DS-2019-02: **Jort Bergfeld**
Quantum logics for expressing and proving the correctness of quantum programs

ILLC DS-2019-03: **András Gilyén**
Quantum Singular Value Transformation & Its Algorithmic Applications

ILLC DS-2019-04: **Lorenzo Galeotti**
The theory of the generalised real numbers and other topics in logic

ILLC DS-2019-05: **Nadine Theiler**
Taking a unified perspective: Resolutions and highlighting in the semantics of attitudes and particles

ILLC DS-2019-06: **Peter T.S. van der Gulik**
Considerations in Evolutionary Biochemistry

ILLC DS-2019-07: **Frederik Möllerström Lauridsen**
Cuts and Completions: Algebraic aspects of structural proof theory

ILLC DS-2020-01: **Mostafa Dehghani**
Learning with Imperfect Supervision for Language Understanding

ILLC DS-2020-02: **Koen Groenland**
Quantum protocols for few-qubit devices

ILLC DS-2020-03: **Jouke Witteveen**
Parameterized Analysis of Complexity

ILLC DS-2020-04: **Joran van Apeldoorn**
A Quantum View on Convex Optimization

ILLC DS-2020-05: **Tom Bannink**
Quantum and stochastic processes

ILLC DS-2020-06: **Dieuwke Hupkes**
Hierarchy and interpretability in neural models of language processing

ILLC DS-2020-07: **Ana Lucia Vargas Sandoval**
On the Path to the Truth: Logical & Computational Aspects of Learning

ILLC DS-2020-08: **Philip Schulz**
Latent Variable Models for Machine Translation and How to Learn Them

ILLC DS-2020-09: **Jasmijn Bastings**
A Tale of Two Sequences: Interpretable and Linguistically-Informed Deep Learning for Natural Language Processing

ILLC DS-2020-10: **Arnold Kochari**
Perceiving and communicating magnitudes: Behavioral and electrophysiological studies

ILLC DS-2020-11: **Marco Del Tredici**
Linguistic Variation in Online Communities: A Computational Perspective

ILLC DS-2020-12: **Bastiaan van der Weij**
Experienced listeners: Modeling the influence of long-term musical exposure on rhythm perception

ILLC DS-2020-13: **Thom van Gessel**
Questions in Context

ILLC DS-2020-14: **Gianluca Grilletti**
Questions & Quantification: A study of first order inquisitive logic

ILLC DS-2020-15: **Tom Schoonen**
Tales of Similarity and Imagination. A modest epistemology of possibility

ILLC DS-2020-16: **Ilaria Canavotto**
Where Responsibility Takes You: Logics of Agency, Counterfactuals and Norms

ILLC DS-2021-01: **Yfke Dulek**
Delegated and Distributed Quantum Computation

ILLC DS-2021-02: **Elbert J.Booij**
The Things Before Us: On What it Is to Be an Object

ILLC DS-2021-03: **Seyyed Hadi Hashemi**
Modeling Users Interacting with Smart Devices

Titles in the SIKS Dissertation Series:

2018 05 Hugo Huurdeman (UVA), Supporting the Complex Dynamics of the Information Seeking Process
06 Dan Ionita (UT), Model-Driven Information Security Risk Assessment of Socio-Technical Systems
07 Jieting Luo (UU), A formal account of opportunism in multi-agent systems
08 Rick Smetsers (RUN), Advances in Model Learning for Software Systems
09 Xu Xie (TUD), Data Assimilation in Discrete Event Simulations
10 Julienka Mollee (VUA), Moving forward: supporting physical activity behavior change through intelligent technology
11 Mahdi Sargolzaei (UVA), Enabling Framework for Service-oriented Collaborative Networks
12 Xixi Lu (TUE), Using behavioral context in process mining
13 Seyed Amin Tabatabaei (VUA), Computing a Sustainable Future
14 Bart Joosten (UVT), Detecting Social Signals with Spatiotemporal Gabor Filters
15 Naser Davarzani (UM), Biomarker discovery in heart failure
16 Jaebok Kim (UT), Automatic recognition of engagement and emotion in a group of children
17 Jianpeng Zhang (TUE), On Graph Sample Clustering
18 Henriette Nakad (UL), De Notaris en Private Rechtspraak
19 Minh Duc Pham (VUA), Emergent relational schemas for RDF
20 Manxia Liu (RUN), Time and Bayesian Networks
21 Aad Slootmaker (OUN), EMERGO: a generic platform for authoring and playing scenario-based serious games
22 Eric Fernandes de Mello Araujo (VUA), Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks
23 Kim Schouten (EUR), Semantics-driven Aspect-Based Sentiment Analysis
24 Jered Vroon (UT), Responsive Social Positioning Behaviour for Semi-Autonomous Telepresence Robots
25 Riste Gligorov (VUA), Serious Games in Audio-Visual Collections
26 Roelof Anne Jelle de Vries (UT),Theory-Based and Tailor-Made: Motivational Messages for Behavior Change Technology
27 Maikel Leemans (TUE), Hierarchical Process Mining for Scalable Software Analysis
28 Christian Willemse (UT), Social Touch Technologies: How they feel and how they make you feel
29 Yu Gu (UVT), Emotion Recognition from Mandarin Speech
30 Wouter Beek, The "K" in "semantic web" stands for "knowledge": scaling semantics to the web

2019 01 Rob van Eijk (UL),Web privacy measurement in real-time bidding systems. A graph-based approach to RTB system classification

02 Emmanuelle Beauxis Aussalet (CWI, UU), Statistics and Visualizations for Assessing Class Size Uncertainty
03 Eduardo Gonzalez Lopez de Murillas (TUE), Process Mining on Databases: Extracting Event Data from Real Life Data Sources
04 Ridho Rahmadi (RUN), Finding stable causal structures from clinical data
05 Sebastiaan van Zelst (TUE), Process Mining with Streaming Data
06 Chris Dijkshoorn (VU), Nichesourcing for Improving Access to Linked Cultural Heritage Datasets
07 Soude Fazeli (TUD), Recommender Systems in Social Learning Platforms
08 Frits de Nijs (TUD), Resource-constrained Multi-agent Markov Decision Processes
09 Fahimeh Alizadeh Moghaddam (UVA), Self-adaptation for energy efficiency in software systems
10 Qing Chuan Ye (EUR), Multi-objective Optimization Methods for Allocation and Prediction
11 Yue Zhao (TUD), Learning Analytics Technology to Understand Learner Behavioral Engagement in MOOCs
12 Jacqueline Heinerman (VU), Better Together
13 Guanliang Chen (TUD), MOOC Analytics: Learner Modeling and Content Generation
14 Daniel Davis (TUD), Large-Scale Learning Analytics: Modeling Learner Behavior & Improving Learning Outcomes in Massive Open Online Courses
15 Erwin Walraven (TUD), Planning under Uncertainty in Constrained and Partially Observable Environments
16 Guangming Li (TUE), Process Mining based on Object-Centric Behavioral Constraint (OCBC) Models
17 Ali Hurriyetoglu (RUN),Extracting actionable information from microtexts
18 Gerard Wagenaar (UU), Artefacts in Agile Team Communication
19 Vincent Koeman (TUD), Tools for Developing Cognitive Agents
20 Chide Groenouwe (UU), Fostering technically augmented human collective intelligence
21 Cong Liu (TUE), Software Data Analytics: Architectural Model Discovery and Design Pattern Detection
22 Martin van den Berg (VU),Improving IT Decisions with Enterprise Architecture
23 Qin Liu (TUD), Intelligent Control Systems: Learning, Interpreting, Verification
24 Anca Dumitrache (VU), Truth in Disagreement - Crowdsourcing Labeled Data for Natural Language Processing
25 Emiel van Miltenburg (VU), Pragmatic factors in (automatic) image description
26 Prince Singh (UT), An Integration Platform for Synchromodal Transport

27 Alessandra Antonaci (OUN), The Gamification Design Process applied to (Massive) Open Online Courses
28 Esther Kuindersma (UL), Cleared for take-off: Game-based learning to prepare airline pilots for critical situations
29 Daniel Formolo (VU), Using virtual agents for simulation and training of social skills in safety-critical circumstances
30 Vahid Yazdanpanah (UT), Multiagent Industrial Symbiosis Systems
31 Milan Jelisavcic (VU), Alive and Kicking: Baby Steps in Robotics
32 Chiara Sironi (UM), Monte-Carlo Tree Search for Artificial General Intelligence in Games
33 Anil Yaman (TUE), Evolution of Biologically Inspired Learning in Artificial Neural Networks
34 Negar Ahmadi (TUE), EEG Microstate and Functional Brain Network Features for Classification of Epilepsy and PNES
35 Lisa Facey-Shaw (OUN), Gamification with digital badges in learning programming
36 Kevin Ackermans (OUN), Designing Video-Enhanced Rubrics to Master Complex Skills
37 Jian Fang (TUD), Database Acceleration on FPGAs
38 Akos Kadar (OUN), Learning visually grounded and multilingual representations

2020 01 Armon Toubman (UL), Calculated Moves: Generating Air Combat Behaviour
02 Marcos de Paula Bueno (UL), Unraveling Temporal Processes using Probabilistic Graphical Models
03 Mostafa Deghani (UvA), Learning with Imperfect Supervision for Language Understanding
04 Maarten van Gompel (RUN), Context as Linguistic Bridges
05 Yulong Pei (TUE), On local and global structure mining
06 Preethu Rose Anish (UT), Stimulation Architectural Thinking during Requirements Elicitation - An Approach and Tool Support
07 Wim van der Vegt (OUN), Towards a software architecture for reusable game components
08 Ali Mirsoleimani (UL),Structured Parallel Programming for Monte Carlo Tree Search
09 Myriam Traub (UU), Measuring Tool Bias and Improving Data Quality for Digital Humanities Research
10 Alifah Syamsiyah (TUE), In-database Preprocessing for Process Mining
11 Sepideh Mesbah (TUD), Semantic-Enhanced Training Data AugmentationMethods for Long-Tail Entity Recognition Models
12 Ward van Breda (VU), Predictive Modeling in E-Mental Health: Exploring Applicability in Personalised Depression Treatment
13 Marco Virgolin (CWI), Design and Application of Gene-pool Optimal Mixing Evolutionary Algorithms for Genetic Programming

14 Mark Raasveldt (CWI/UL), Integrating Analytics with Relational Databases
15 Konstantinos Georgiadis (OUN), Smart CAT: Machine Learning for Configurable Assessments in Serious Games
16 Ilona Wilmont (RUN), Cognitive Aspects of Conceptual Modelling
17 Daniele Di Mitri (OUN), The Multimodal Tutor: Adaptive Feedback from Multimodal Experiences
18 Georgios Methenitis (TUD), Agent Interactions & Mechanisms in Markets with Uncertainties: Electricity Markets in Renewable Energy Systems
19 Guido van Capelleveen (UT), Industrial Symbiosis Recommender Systems
20 Albert Hankel (VU), Embedding Green ICT Maturity in Organisations
21 Karine da Silva Miras de Araujo (VU), Where is the robot?: Life as it could be
22 Maryam Masoud Khamis (RUN), Understanding complex systems implementation through a modeling approach: the case of e-government in Zanzibar
23 Rianne Conijn (UT), The Keys to Writing: A writing analytics approach to studying writing processes using keystroke logging
24 Lenin da Nobrega Medeiros (VUA/RUN), How are you feeling, human? Towards emotionally supportive chatbots
25 Xin Du (TUE), The Uncertainty in Exceptional Model Mining
26 Krzysztof Leszek Sadowski (UU), GAMBIT: Genetic Algorithm for Model-Based mixed-Integer opTimization
27 Ekaterina Muravyeva (TUD), Personal data and informed consent in an educational context
28 Bibeg Limbu (TUD), Multimodal interaction for deliberate practice: Training complex skills with augmented reality
29 Ioan Gabriel Bucur (RUN), Being Bayesian about Causal Inference
30 Bob Zadok Blok (UL), Creatief, Creatieve, Creatiefst
31 Gongjin Lan (VU), Learning better – From Baby to Better
32 Jason Rhuggenaath (TUE), Revenue management in online markets: pricing and online advertising
33 Rick Gilsing (TUE), Supporting service-dominant business model evaluation in the context of business model innovation
34 Anna Bon (MU), Intervention or Collaboration? Redesigning Information and Communication Technologies for Development
35 Siamak Farshidi (UU), Multi-Criteria Decision-Making in Software Production

2021 01 Francisco Xavier Dos Santos Fonseca (TUD),Location-based Games for Social Interaction in Public Space
02 Rijk Mercuur (TUD), Simulating Human Routines:Integrating Social Practice Theory in Agent-Based Models
03 Seyyed Hadi Hashemi (UvA), Modeling Users Interacting with Smart Devices