LUDWIG-MAXIMILIANS-UNIVERSITÄT AT MÜNCHEN
Department "Institut für Informatik"
Lehr- und Forschungseinheit Medieninformatik
Prof. Dr. Heinrich Hußmann

Masterarbeit

Understanding and Predicting Web Browsing Behavior

Changkun Ou
hi@changkun.us

Bearbeitungszeitraum:	12.8.2018 bis 12.2.2019
Betreuer:	Malin Eiband and Dr. Daniel Buschek
Verantw. Hochschullehrer:	Prof. Dr. Heinrich Hußmann

Aufgabenstellung

Understanding and Predicting User Browsing Behavior

Problem Statement Understanding user behavior helps the designer optimize their product user experiences. Meanwhile, the user can benefit from a better product. Since user intents are elusive, changeable and sometimes even undetermined, predict their behavior usually tricky and impossible. In most cases, a user may perform a series of wasted actions before reach an intended destination. Nevertheless, user intents become clear step by step after performs a series of actions in a given context.

Scope of the Thesis Applying machine learning approaches in human-computer interaction gains its popularity. The objective of this thesis is to use machine learning to develop techniques that help a user better understand the user's behavior and benefits from it. As a first step, a literature review is needed for finding research questions in a specific research field. Based on the literature review, an interpretable machine learning model should be developed. Afterward, the thesis should design a reasonable experiment with a suitable theoretical foundation for a user study of the developed model, and give a comprehensive analysis of the developed model. A demonstration software may be developed for explaining the model application.

Tasks (1) Conduct a literature review to identify research questions regarding a specific field of research that are of interest to researchers and practitioners. (2) Design a machine learning model and design an appropriate experiment with theoretical support to justify model performance. A comprehensive analysis is needed for evaluating its interpretability. (3) Develop an application as a demonstration of the model.

Requirements Profound knowledge in human-computer interaction, strong skills in mathematical modeling and machine learning approaches, independent scientific work and creative problem solving, industrial experience in web development, and architecting.

Keywords Clickstream, User Browsing Behavior, Machine Learning, Web

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig angefertigt, alle Zitate als solche kenntlich gemacht sowie alle benutzten Quellen und Hilfsmittel angegeben habe.

München, February 8, 2019 . . . .

Acknowledgments

I would like to appreciate Malin Eiband and Dr. Daniel Buschek for many of their great and constructive discussions, suggestion around thesis topic selection, model statements, thesis structuring as well as their pacient of given supervision to the thesis. The thesis would not be accomplished without them. Afterward, I am full of gratitude to Prof. Hussmann and Prof. Butz for offering me an unexpected opportunity to study aboard in Germany as well as their kindly helps regarding study and living in my academic period. Next, I wish to thank my best friend Yi Xia, schoolfellow Yifei Zhan and Xingying Chen, former colleague Jin Luo, and prior supervisor Yingding Wang in a Master's course "Human Computation" for their inspiring discussions, proofreading, and giving feedbacks surrounding the contents of my thesis. Finally, I will forever be beholden to my parents for their continuous supports while my postgraduate study, since they dedicated their live to my better future.

Abstract

Clickstream applications appeared at the end of last century and have proliferated the heart of our Internet world. Trades, public opinions, and almost every web requests are precisely recorded on server-side log files. The fundamental interaction between a web service client and server stands immutably, even though mobile devices have governed our daily life. This thesis proposes a machine learning model that characterizes user browsing behavior while including multi-tab branching and backtrack actions in a browser instead of web request-based clickstreams. The model is named the Action Path model (APM). To justify the APM, a lab study is established and individuals' clickstream data is collected, which consisted of chronologic URLs and corresponding stay durations for each URL. The thesis designed nine different contexts given web browsing tasks for three mainstream websites based on the theory of information behavior. Each website has three types of tasks: a goal-oriented task, fuzzy task and exploring browsing task. They characterize the corresponding three browsing behaviors. The thesis seeks to achieve the following goals by analyzing the subject's trace from a lab study: 1) Understanding: identify if browsing behaviors are distinguishable and find common patterns that appear in an action path. 2) Classification: separate and report browsing behaviors on the web, which will help users to better understand their status. 3) Prediction: present the future click path in more than one step with the given context of the browsing history in a session. The quantitative analysis in this thesis indicates that goal-oriented, fuzzy, and exploring browsing behaviors are classifiable with 100% accuracy based on the combination of chronologic URLs and stay duration. The prediction performance of APM indicates higher than 60% accuracy for three to five steps of future clickstream prediction. The qualitative analysis of the APM indicates five observed patterns, including "ring","star", "overlap", "hesitation" and "cluster" patterns, which represent the patterns of an action path. To illustrate the application, a browser plugin is developed that proactively serves users, and suggests predictions for the possible future user clicks. Furthermore, the thesis discusses a generalized design of APM and plugin communication protocol. This discussion explores the possibility of formalizing the model and protocol as standard Web APIs to help designers and developers to improve and monitor the user experience of their products. To the best knowledge of the author of the thesis, the proposed APM is the first model with a detailed study regarding web browsing behavior modeling based on clickstreams collected from the client side.

Contents

1 Introduction

> That men do not learn very much from the lessons of history is the most important of all the lessons that history has to teach.
>
> Aldous Huxley

1.1 History of Clickstream Research

The notion "clickstream" [Friedman, Wayne and Weaver, Jane, 1995] was first coined in 1995 when a media article introduced the novel concept of tracing the cyberlife of users over what is currently known as the "Internet". "clickstream" contains a sequence of hyperlinks clicked by a website user over time. In 1995, the most popular server software Apache HTTP [The Apache Software Foundation, 1995] proxy on the Web was developed with a feature that recorded the access log of entries. Afterwards, people realized the potential danger and value of tracing cyberspace. A major discussion was concluded over clickstream issues, such as the frequency-based mining of clickstream [Brodwin, D., D. O'Connell, and M. Valdmanis., 1995], privacy concerns [Reidenberg, 1996], and the database schema of session-based time series data [Courtheoux, 2000].

The privacy discussion concluded that collecting traces over the internet clearly violates the rights of users and, breaches the openness and transparency of a service. Serious criticism arose that traces damage democratic governance [Gindin, 1997].

Technology is not guilty. Years of discussion have produced, rules [Reidenberg, 1996] and regulations [Skok, 1999] in cyberspace, means of protecting information privacy in cyberspace transactions [Kang, 1997], and approaches to resolve conflicting issues in international data privacy [Reidenberg, 1999].

Meanwhile, bussinessmen have agilely reasponded to the concept of collecting clickstream and immediately initiated commercial tracking of their customers to measure product success [Schonberg et al., 2000] and improve marketing effects [Novick, Bob, 1995], customer service, and precise advertisements [Reagle and Cranor, 1999, Bucklin and Sismeiro, 2000].

At the turn of this century, there has been common acceptance of the technology of clickstream. Clickstream data has been confirmed by industrial practice, which has opened up a new era in customer service [Walsh, John and Godfrey, Sue, 2000]. Most websites' users habe begun to accept that their click path data will be aggregated and analysed on the server side [Carr, 2000].

Clickstream data grows and becomes plentiful quickly. Researchers have begun to convey track customer selections, which it the original idea behind of clickstream, into various applications, such as usability testing [Waterson et al., 2002a] and understanding social network sentiment [Schneider et al., 2009]. Researchers have also developed visualizing techniques to better interpret clickstream data [Waterson et al., 2002b].

Analysis, reports, and characterizing of clickstream have gained in popularity. Mobasher et al. [Mobasher et al., 2001] have suggested personalizing users based on association rules from their web usage data. Chatterjee et al. [Chatterjee, Patrali and Hoffman, Donna L and Novak, Thomas P, 2003] have first proposed that e-commerce websites should use clickstream instead of essential choice to track customer navigation patterns, thereby associating and binding products for the observing responses of a customer.

As characterization and the understanding of behavior based on clickstream data have become popular, more research have proposed methods to understand server clickstream data. Padmanabhan et al. [Padmanabhan et al., 2001] have proposed an algorithm to address personalization from incomplete clickstream data, which implies the security problem potential of a potential information leak from clickstream data. Regarding search engine indexing, Lourenco at al. [Lourenço and Belo, 2006] recommends an approach for the detection and containment of web crawlers based on server-side recorded visiting log files.

A short review of clickstream history has indicated that almost all research have formulated their methodology based on server-recorded clickstream data. A daily user is always allowed simultaneous accesses to parallel pages and windows and may even be allowed, to switch across multiple websites for a browsing purpose. An obvious missing aspect of those papers is that server-recorded data tends to be incomplete for characterizing a visiting user, and the log data can only applied to on a specific website. The research thesis no longer serves the server-side clickstream; instead, it focuses and contributes to client-side collected clickstream data for a real visiting session of a user in a browser.

1.2 This Thesis

The main part of the thesis is structured in different chapters, and answers the following three research questions:

1. **Understanding**: Why does collecting clickstream on the client-side differ from on the server-side? What are the most significant and identifiable user behaviors and activity patterns that can be observed or algorithmically detected in the context of web browsing that indicate information needs? Which form of quantitative data and what quantitative measures derivable from the client-side clickstream to distinguish or define the different browsing behaviors of a user.

2. **Classification**: How accurately or affirmatively can we progressively model or identify the proposed browsing behaviors that makes an intelligent system can serve a user proactively without requiring the user inquire the system?

3. **Prediction**: How many future movements of a user can be accurately inferred from the context of web browsing, and how much context is required for the prediction?

Firstly, the existing existing user behavior research based on clickstream data is discussed in Chapter 2. Next, the chapter discusses the evolution of the theory regarding information seeking behavior as the experiment's foundation. In addition, the chapter summarizes the reason for the recent increase of the neural approach in different scientific area and the state-of-the-art direction for sequence learning, that have been proposed in neural network research. In Chapter 3, several handcrafted features are defined for a clickstream, such as completion efficiency. Next, the thesis formalize the proposed sequence to sequence encoder and decoder *Action Path model (APM)* for client-side clickstreams as well as the training techniques for the proposed APM. In Chapter 4, the experiment design is presented for the lab study held in this thesis. Based on information behavior theory, the chapter construes the design justification of context-given web browsing tasks for the subjects recruited in the thesis. In Chapter 5, a quantitative analysis with described data from the lab study is conducted, based on support vector machine (SVM), t-SNE, and the proposed APM. The evaluation produces up to 100% of accuracy in classification task. Moreover, the chapter visualizes the clickstream through a directed graph by combining the training model outputs, and it also performs a qualitative analysis on all clickstreams, and the analysis provides evidence that further verifies the correctness of APM. Chapter 6 describes how a browser plugin is developed for Google Chrome as a possible application for APM. The plugin can accurately predict the next possible visiting pages of a user. In addition, the chapter generalizes the design of the plugin's architecture between the client and server. Furthermore, the possibilities of using the architecture as a standard web API for web developers are discussed.

In the last two chapters, the decisions, limitations, and the findings of the thesis are discussed for possible future improvements and directions.

2 Related Works

If I have seen further it is by
standing on the shoulders of Giants.

Isaac Newton

The previous research relating to this thesis is highlighted in this chapter. These research are the existing approaches to clickstream behavior modeling, the evolution of information behavior theory regarding and how it adapts to our digital world, and the most relevant recent advances regarding sequence learning.

2.1 Clickstream Behavior Modeling

2.1.1 Server-side Clickstream Models

Clickstream behavior research can be traced back to the year when the word "clickstream" was invented. Early clickstream behavior research has studied the navigational behavior of user [Mandese, 1995, Brodwin, D., D. O'Connell, and M. Valdmanis., 1995]. These research conducted binary classifications of clickstream based on the degree of linearity.

Mobasher et al. have discovered the effective and scalable techniques [Mobasher et al., 2001] for web personalization by using association rules and built a recommendation system. Goldfrab has investigated [Goldfarb, 2002] the website choice behavior based on clickstream data and has suggested that clickstreams simulate company strategy changes. Chatterjee et al. [Chatterjee, Patrali and Hoffman, Donna L and Novak, Thomas P, 2003] where the first to conduct research on clickstream with regard to an actual commercial website, and they found implies that dynamic advertising based on customer clickstream history influences the future clickstream of the customer and increases interactions with dynamic advertisement. More technically, Ting et al. have used common sequences to determine unexpected browsing behavior [Ting et al., 2005] and used their findings to improve website design.

The most recent research has evolved the approaches of clickstream modeling, Wang et al. [Wang et al., 2016] have proposed an unsupervised approach to model clickstream without labeling. Chi et al. have proposed an analysis framework [Chi et al., 2017] for the general understanding of online information behavior in a specific page. However, their framework only fits for server-side collected clickstreams instead of a real user clickstream.

Then, Wang et al. [Wang et al., 2017] have improved their unsupervised approach and summarized more comprehensive reviews of the existing approaches, such as common subsequences of clickstreams and graph clustering based classification for clickstream behavior modeling that identifies spam and abuse for a specific website. Based on Poisson process, Park et al. have modeled and detected a behavior change among students while learning [Park et al., 2017] to help improve their online learning experience. Amo et al. [Amo Filvá et al., 2018] have further visualized search-stream behavior that serves student clickstream data from a class, and Shimada et al. have proved [Shimada et al., 2018] that online change detection while monitoring student behavior is possible based on a sliding window.

Zaloudek has conducted a review of review on the comparison [Zaloudek, 2018] traditional method to model clickstream data, and proposed a principle component analysis-based method for the semi-supervised learning of clickstream data. However their approach does not work well for clustering task, and the best performance is obtained by a traditional multilayer perceptron algorithm. Chandramohan and Ravindran have further investigated the neural approach to clickstream mining [N and Ravindran, 2018], and verified that based on a server-side collected clickstream, a complex recurrent unit with an attention mechanism can capture whether a user has the intent to buy a product. Surprisingly, Gundala and Spezzano [Gundala and Spezzano, 2018] have used a

Lasso regression based on sophisticated feature engineering with an archived area under the curve score 0.769 for reader demand hyperlink prediction on a Wikipedia clickstream dataset.

2.1.2 Client-side Clickstream Models

Kammenhuber et al. have produced the first study regarding a client-side clickstream [Kammenhuber et al., 2006]. They have proposed a finite-state Markov model that models user's search behavior on a level of topic categories. Unfortunately, their dataset was collected from network package traffic, they did not consider the time and actions that a user spends and made on each page.

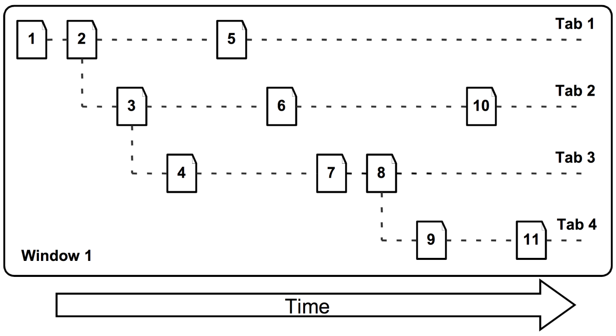

Figure 2.1: Parallel browsing behavior: branching phenomenon [Huang and White, 2010]

Liu et al. [Liu et al., 2010] have studied specific user behavior on dwell time on web pages and concluded that Weibull distribution is the most appropriate distribution for characterizing this behavior. Huang et al. [Huang and White, 2010, Huang et al., 2012] have further noticed the behavior of branching parallel browsing and backtracking browsing behavior on modern browsers, as depicted in Figure 2.1. The authors have also presented a frequent analysis for the individual distribution of these two types of behavior.

The existing research regarding clickstream behavior modeling is either server-side modeling for an individual website or is individually modelized for client-side behaviors with limited information regarding clickstream, which do not accurately represent the ground truths of user behavior. In any case, the existing approaches are based on self-constructed features, the property of Markov memoryless, and so on. Though the most recent approaches use neural networks, their findings only applies to specific context. From the point of view of user behavior, these previous approaches neither unambiguously justify the foundation of their model, nor enable a major performance improvement of their model.

In this thesis, the client-side chronologic URL sequences are serialized with combinations of all these individually studied phenomena, including the branching and backtracking browser feature. These chronologic URLs are used to understand and model the essential user behavior patterns while browsing on the web.

2.2 Sequence to Sequence Learning

Sequence learning is a large scope of research and has been applied to many fields such as typical application machine translation in nature language processing.

Recurrent neural network (RNN) have been described by Werbos [Werbos, 1990] and Rumelhart et al. [Rumelhart et al., 1988], the original RNN generalizes feedforward neural networks for sequence based data.

Given a sequence of input $(i_1, i_2, ..., i_T)$, the original RNN computes a sequence of outputs $(o_1, o_2, ..., o_T)$ by iterating the activation function Equation 1:

$$
\begin{aligned}
k_t &= \sigma\left(W_{hi} i_t + W_{hh} k_{t-1}\right) \\
o_t &= W_{oh} k_t, t = 1, 2, ..., T
\end{aligned} \tag{1}
$$

where $\sigma(x) = \frac{1}{1+\exp\{-x\}}$ is a non-linear transformation function, and W_{oh}, W_{hh}, W_{hi} are weight parameters between output, hidden and input layers, as shown in Figure 2.2.

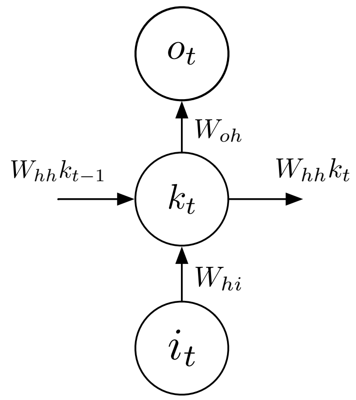

Figure 2.2: Vanilla recurrent unit: the original RNN uses linear weights transformation and a σ non-linear transformation between inputs and outputs as a recurrent unit

The most widely used recurrent units in RNN are the Long-Short-Term memory (LSTM) unit [Hochreiter and Schmidhuber, 1997] or gated recurrent unit (GRU) [Cho et al., 2014], These units provide a performance than is significantly superior to traditional hidden Markov models in machine translation [Garg and Agarwal, 2019].

The LSTM unit has a context cell and three regulators: input gate, output gate and forget gate. The context cell maintains dependencies between inputs of the unit as a form of long term memory. The input gate takes the historical hidden state as well as the current input and controls the input value to the recurrent unit. The output gate is responsible for the control of output activations. The forget gate resets and decides retaining values of the recurrent unit as a form of short-term memory. Similarly, the GRU simplifies the structure of the LSTM into an update gate and a reset gate.

On the other hand, the vanilla RNN transfers and maps a sequence to another sequence if the inputs and the outputs are aligned with equal length. Apparently, the major constraint of the vanilla RNN is that the model cannot address a problem if the inputs and outputs are provided in different lengths with complicated and non-monotonic relationships.

Stutskever et al. [Sutskever et al., 2014] have presented a general end-to-end approach to sequence learning models in machine translation that estimates the conditional probability of $p(o_1, o_2, ..., o_{T'} | i_1, i_2, ..., i_T)$ where $(i_1, i_2, ..., i_T)$ is an input sequence, $(o_1, o_2, ..., o_{T'})$ is a corresponding output sequence, and T does not have to equal with T'.

In machine translation, a series of words are considered to be a sequence of vectors, and neural network-based models are considered to be representative of the learning of nature languages. The initial vectors of word were one-hot encoded vectors and received updates over the training and learning.

The recent advances of representation learning uses a distributed representation of the word2vec model [Mikolov et al., 2013a], which achieves better performance in natural language processing. The word2vec model introduced the continuous bag-of-word model and the skip-gram model as efficient methods for learning the high-quality vector representation of words. The bag-of-word model is faster and the skip-gram model is slower but achieves better performance for infrequent words.

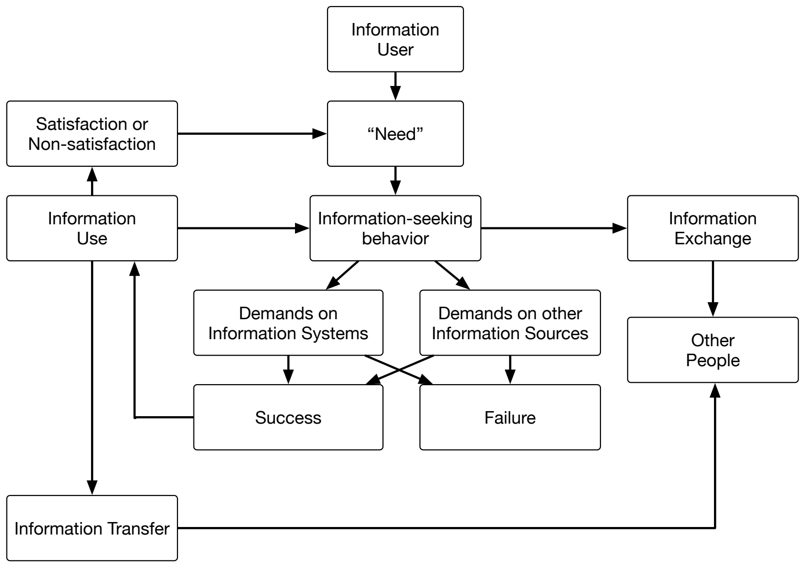

Figure 2.3: Wilson's information seeking behavior model [Wilson, 1981]

2.3 Theory of Information Behavior on the Web

The thesis relates to information behavior theory because it supports the foundation of our user study. This subsection discusses how the theory was concluded and how the principles of the theory that sustain the thesis.

Information behavior research encompasses intentional information seeking and unintentional information encounters. The roots to information behavior theory relates to information needs and uses [Fisher and Julien, 2009] that arose in the 1960s.

However, the concept of information seeking behavior, was coined in the late 1981s by Thomas Wilson [Wilson, 1981]. He tries to formalize the process or activities of a conscious effort regarding information needs and uses. Figure 2.3 illustrates the model of information behavior that was proposed.

Wilson's model has been used for many years since its inception, and has been revised and adapted to our digital world because the digital systems learn user preferences and change [Giannini, 1998] the way we receive information.

David Ellis has described a detailed group of activities for information seeking behavior [Ellis, 1989] and applied it to the industrial as well as physical and social science [Ellis et al., 1993] environments [Ellis and Haugan, 1997]. His analysis was based on grounded theory approach [Aceto et al., 1994] and semi-structured interviews.

Choo et al. adapts Ellis' Model and discussed [Choo et al., 1999] the information seeking behavior on the web through different activities other than a single process. The activities are: ***starting, chaining, browsing, differentiating, monitoring, and extracting***.

"*Starting*" on the web indicates that a user identifies websites or pages that contain the information regarding their interests. "*Chaining*" indicates that a user follows the starting page to other related pages. "*Browsing*" represents the activity that a user engages in when they skim on the web and quickly view the top-level information. "*differentiating*" describes how a user on the web selects useful pages and choose differentiated targets. "*Monitoring*" activity is used for receiving updates on the sites or for revisiting the previously visited pages. Finally, "*extracting*" refers to a user systematically extracting information from an interested page or website while browsing.

By applying these activities, Choo et al. [Choo et al., 1999] have concluded that the general user behaviors on the web are undirected viewing, conditioned viewing, informal search and formal search. Johnson has described further [Johnson, Ross, 2017] for seven detailed behaviors patterns on the web but did not provide a working study that verified or proved their formation.

Although Wilson's model and Ellis' model are revised in recent works, these improvements are more generic and are too complex for describing user information behavior on the web, which cannot adapts to the experiment design (discuss detailly in Chapter 4 and Section 7.2). This thesis uses an antecedent of Wilson's framework [Wilson, 1997] and Ellis' model [Ellis and Haugan, 1997] to formalize and justify the lab study in Chapter 4. This experiment form the foundation of this work.

3 Action Path Models

> It is impossible to separate a cube into two cubes, or a fourth power into two fourth powers, or in general, any power higher than the second, into two like powers. I have discovered a truly marvelous proof of this, which this margin is too narrow to contain.
>
> Pierre de Fermat

In this chapter, few concepts and metrics are formalized for clickstream data, and then **a proposed clickstream model named *Action Path model (APM)*** is described based on a recurrent neural network that models client-side web browsing behavior. An *action path* is different than the original clickstream concept since a user may *switch browser tabs* for parallel viewing [Huang and White, 2010] or uses *back button* for backtracking viewing [Huang et al., 2012] as discussed in Chapter 2, namely, **an action path contains a series of browsing action performed by a user**. Server-side collected clickstream does not contain such detailed level of user clickstream. The term action path is a generalized concept of clickstream, which replaces individual URLs to chronological ordered user actions (with back button and browser tab switch effects) in a browser. Figure 3.1 illustrates a simplified version of an action path that compares vanilla clickstream.

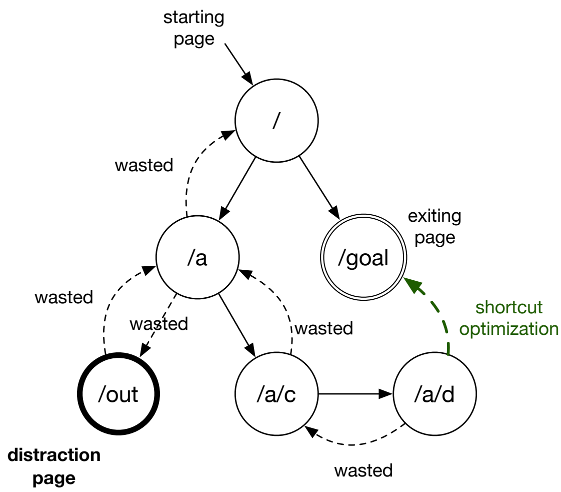

Figure 3.1: A simple action path. A user starts from the starting page, and performed a series of page click actions, ends on a exiting page. The server side records clickstream in the following order: / → /a → /out → /a/c → /a/d → /goal. However the actual user actions are: / → /a → /out → /a → /a/c → /a/d → /a/c → /a → / → /goal. The records from server side lost the interaction details between users and browsers. Node that /out is a distraction page in the graph, which may located in a different website (e.g. advertisement), and black dashed arrows are wasted user actions. The /goal page may not clear in the beginning of the clickstream, one can generate a shortcut optimization navigation to the /goal page while more clickstream context be presented, i.e. an optimized user action is / → /a → /a/c → /a/d → /goal. In this case, the demand page of the visit session is discovered in /a/d.

For the convenience of discussion, **the thesis mix the use of term *action path* and *clickstream* to indicate a chronologically ordered user actions in web browsing**.

3.1 Completion Efficiency

An action path of a visiting session starts from a starting page and ends on an existing page. Since the thesis consider the effect of browser back button and browser tab switches, a previous page could easily be visited twice, if a user clicked the back button. Therefore, a page may direct to multiple pages. *For instance, an action path can degrade to a linked list if the user clicks through different pages without using the back button and switching tabs; or an action path can become a 1-to-n bipartite graph if a user use back button back to the previous page after clicking a page or only switching tabs from a specific page to one another*, as shown in Figure 3.2.

As a result, The definition of *completion efficiency* is based on shortest path from starting page to exiting page, and stay duration is an arc weight that represents the number of seconds that a user spend on an arc tail in an action path.

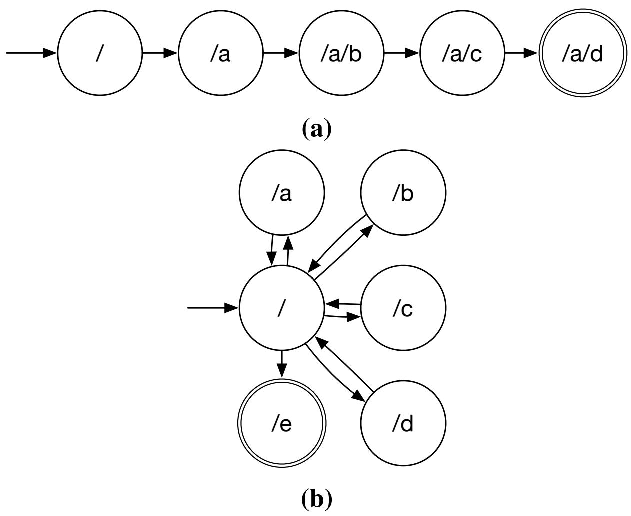

Figure 3.2: Two particular case of an action path: an action path that degrade to a linked list if the user click through different pages without using back button and switching tabs (3.2a), and an action path that represented in 1-to-n bipartite graph if a user use back button back to the previous page after clicking a page or only switching tabs from a specific page to one another (3.2b).

Let a directed cyclic graph represents an action path, each node represents a visited page, and each arc has a weight that represents the study duration of its tail node. Assume the total stay duration of the shortest path from the starting page to the existing page is d_s, and the total stay duration of the action path is D, the number of nodes in the shortest path is n_s, the total nodes in an action path is N, the *completion efficiency* E is defined as follows Equation 2:

$$\begin{aligned} E = w_1 \frac{n_s}{N} + w_2 \frac{d_s}{D} \\ w_1 + w_2 = 1 \end{aligned} \tag{2}$$

where w_1, w_2 are hyper-parameters to balancing the importance of action path and stay duration. According to the discussion of two special cases of action path, it is trivial to show the range of E is $(0, 1]$. As a compliment, *zero completion efficiency* is defined if a user cannot complete a clickstream in a browsing session. Therefore the range of E is in $[0, 1]$.

Remark 1 The definition of completion efficiently uses the term of *shortest path*, which is the problem of finding a path between the starting page and exiting page in an action path (consider as directed cyclic graph) such that the sum of the stay duration of its constituent pages is minimized.

The problem can be solved by Dijkstra's [Dijkstra, 1959] shortest-path algorithm. It selects the unvisited nodes with the smallest weights, calculates the distance through it to each unvisited neighbor, then updates the distance of neighbor distance if the distance is smaller than one another. The process is proven that converges to result in the shortest path.

Remark 2 An action path may increases with more nodes (web pages) over time. The starting page of an active path was always the first page when the browser was opened. However, one can always treat the currently visited page is the exiting page due to we do not know when a user will exit browsing overtime at the moment. Consequently, function E is changing over browsing.

Remark 3 The completion efficiency is a feature that handcrafted for a classification task in Section 5.2.1, it is created that intend to uses the structure information (arc weights) of a clickstream.

3.2 *url2vec* Embedding

As discussed in Section 2.2, similar idea is conveyed from *word2vec* model. This thesis proposes a *url2vec* model for client-side clickstream data.

The purpose of *url2vec* model is to construct URL representations that better predict the surrounding URLs in a clickstream. Briefly, given an action path of URLs $\mathrm{URL}_1, \mathrm{URL}_2, ..., \mathrm{URL}_T$, the objective of *url2vec* is to maximize the average logarithm softmax probability:

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le i\le c, i\ne 0}\log p(\mathrm{URL}_{t+i}|\mathrm{URL}_t) \\
p(\mathrm{URL}_{t+i}|\mathrm{URL}_t) &= \frac{\exp\left(v_{\mathrm{URL}_{t+i}}^{\top} v_{\mathrm{URL}_t}\right)}{\sum_{\text{all URLs}}\exp\left(v_{\mathrm{URL}_{t+i}}^{\top} v_{\mathrm{URL}_t}\right)}
\end{aligned}
\tag{3}
$$

where c is the size of embedding context, which is a function of starting page, v_{URL_t} is one-hot encoded representation of input URLs, and $v_{\mathrm{URL}_{t+i}}$ is the vector embedding of output representations.

Remark 1 The model described by Equation 3 is essentially a three layer neural network: input layer of *one-hot* encoded URLs (a group of binaries that a component of a one-hot encoded vector is a representative of a URL under a finite set of existing URLs), a hidden layer of feature representation and an output layer share weights to the learned embeddings of input URLs.

Remark 2 The probability in Equation 3 is impractical due to $\nabla \log p(\mathrm{URL}_{t+i}|\mathrm{URL}_t)$ is large because of exponential terms in softmax, two numerical optimizations [Mikolov et al., 2013b] based on Hofmann Tree and Negative Sampling are proposed by Mikolv.

Remark 3 The probability can also be interpreted from a Bayesian perspective, which provides an intuition of this definition. $p(\mathrm{URL}_{t+i}|\mathrm{URL}_t)$ can be considered as a posterior probability. **Since v_{URL_t} was initialized as a one-hot encoded vector input to the embedding neural network, the item can be treated as a prior, and the denominator is a normalization term.**. Furthermore, the dot product between $v_{\mathrm{URL}_{t+i}}^{\top}$ and v_{URL_t} is a representation of cosine similarity, which represents the closest surrounding URLs in same direction of vectors.

Remark 4 The v_{URL_t} is updated through gradient descent while training from a one-hot encoded sparse high dimensional space to a densely distributed $(v_{\mathrm{URL}_t}, v_{\mathrm{URL}_{t+1}})$ pairs. The pair is the ground truth URL relationships in a browsing behavior.

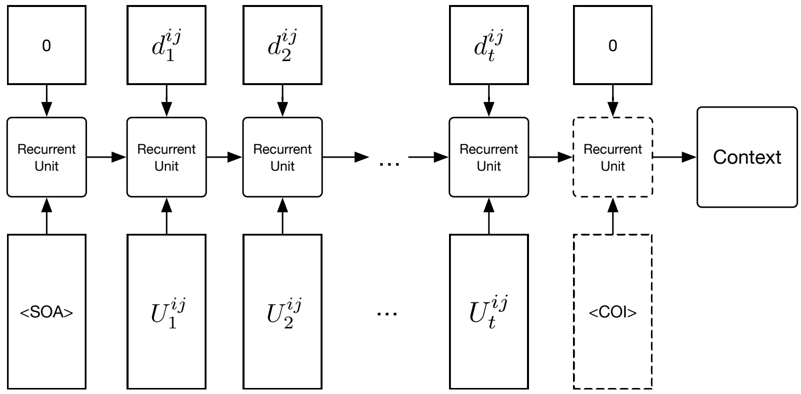

Figure 3.3: An unrolled illustration of context encoder of the APM. In the encoder, a starting mark "<SOA>" is used as a sign of start feeding URLs, and a trigger mark "<COI>" as a sign to trigger decoder to decodes encoded context tensor. The trigger mark is automatically inserted after the k-th URL while training at the end of the encoder model over time, k is increasing over time. Besides, the recurrent unit is not detail described in the figure but afterward.

3.3 Action Path Model

The APM convey a similar idea from Stutskever's sequence to sequence translation as discussed in Section 2.2.

An *action path* from user i in session j consist of a sequence of *url2vec* embedded vectors $(U_1^{ij}, U_2^{ij}, ..., U_n^{ij})$ and a sequence of time duration $(d_1^{ij}, d_2^{ij}, ..., d_n^{ij})$, since each URL has a corresponding number that represents the time duration of a user spent on a given page. The APM consist a context encoder and a context decoder that illustrated in the subsequent subsections.

3.3.1 Context Encoder

Context encoder encodes URLs one by one over timestamp and produces a context tensor that encodes the historical user actions, as shown in Figure 3.3.

In the encoder, a starting mark is practically inserted (a mark is a special URL vector that differ from any other realistic URL one-hot encoded vectors) "<SOA>" (*Start of Action*) as a sign of start feeding URLs to the encoder, and a trigger mark "<COI>" (*Change of Intention*) as a sign to trigger decoder to decodes encoded context tensor.

Note that the input URLs to encoder's recurrent unit are preprocessed through url2vec embeddings, which has learned and updated from one-hot encoded vectors to densely distributed vectors.

3.3.2 Context Decoder

Context decoder decodes the context tensor produced by the encoder into a series of URLs. A practically fed prediction mark "<SOP>" (*Start of Prediction*) is used as a sign to initiate the decoding of encoded context. At the end of decoder, decoder produces an ending mark "<EOA>" (*End of Action*) that terminates the decoding process.

Note that the decoder model in the training phase and prediction phase is different. In the training phase, teacher forcing strategy [Williams and Zipser, 1989] is used, the strategy supplies observed user actions as inputs in the decoder. In the evaluation phase, the decoder uses the output from the recurrent unit as an input, shown through dashed lines in Figure 3.4.

In APM, a decoder outputs vectors first, and it has two strategies in translating vectors to URLs. The first strategy is to use an *arguments of the maxima (argmax)* function to select the

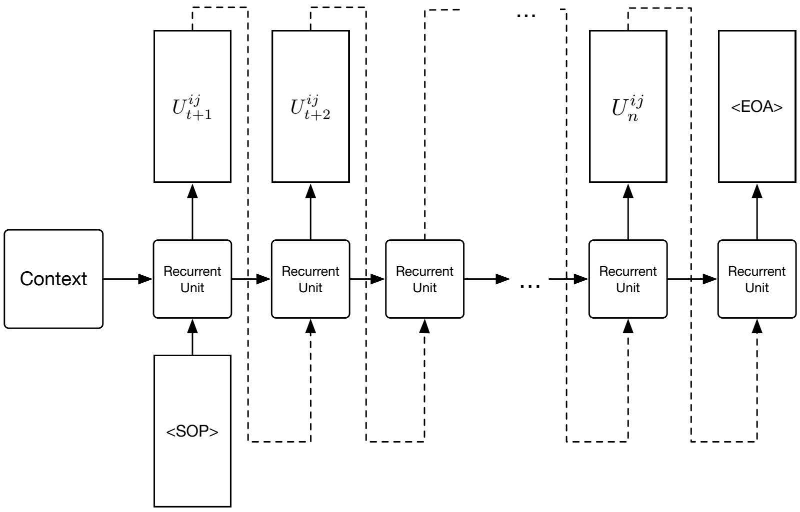

Figure 3.4: The context decoder of the APM. In the decoder, a prediction mark "<SOP>" is used to initiate the decoding process, and an ending mark "<EOA>" as a sign to terminate decode process. The output of the decoder uses a softmax intermediate operation to magnify and normalize the probability of predicted URL embedding. Also, the recurrent unit is not detail described in the figure but afterward.

component with maximum probability of a vector; APM use this strategy for performance evaluation in Section 5.3. Another strategy is to select a series of URLs that gains the highest joint probability, APM use this strategy for action path optimization in Section 3.4.

3.3.3 Recurrent Unit

The recurrent unit in the APM is not as standard as original LSTM or GRU, which is one of the major contributions of the thesis. A recurrent unit that is designed for APM must accept two types of data, including chronological URLs and time stay duration.

When using LSTM-like recurrent unit, APM feeds time duration $(d_1^{ij}, d_2^{ij}, ..., d_n^{ij})$ into input gate I_t, and others (forget gate F_t, output gate O_t, memory cell C_t and hidden state h_t) remains the same:

$$\begin{aligned}
I_t =& \sigma(P^{(I)}U_t^{ij} + Q^{(I)}h_{t-1} + \frac{d_t^{ij}}{d_t^{ij}+1})) \\
F_t =& \sigma(P^{(F)}U_t^{ij} + Q^{(F)}h_{t-1} + b^{(F)}) \\
O_t =& \sigma(P^{(O)}U_t^{ij} + Q^{(O)}h_{t-1}) \\
C_t =& F^{(t)} \circ C_{t-1} + I_t \circ \tanh(P^{(C)}U_t^{ij} + Q^{(C)}h_{t-1}) \\
h_t =& O_t \circ \tanh(C_t)
\end{aligned} \tag{4}$$

where $t = 1, 2, ..., n; P^{(I)}, Q^{(I)}, P^{(F)}, Q^{(F)}, P^{(O)}, Q^{(O)}$ are shared weight parameters, $b^{(F)}$ is a bias in forget gate F_t, $\circ$ represents element-wise product of two matrices.

When using GRU as recurrent unit base, APM feeds time stay duration $(d_1^{ij}, d_2^{ij}, ..., d_n^{ij})$ in to update gate Z_t, and others (reset gate R_t, hidden state h_t) stay the same:

$$
\begin{aligned}
Z_t =& \sigma(P^{(Z)}U_t^{ij} + Q^{(Z)}h_{t-1} + \frac{d_t^{ij}}{d_t^{ij}+1}) \\
R_t =& \sigma(P^{(R)}U_t^{ij} + Q^{(R)}h_{t-1}) \\
h_t =& (1-Z_t) \circ \tanh(P^{(H)}U_t^{ij} + Q^{(H)}h_{t-1}) + Z_t \circ h_{t-1}
\end{aligned}
\tag{5}
$$

where $t = 1,2,...,n; P^{(Z)}, Q^{(Z)}, P^{(R)}, Q^{(R)}, P^{(H)}, Q^{(H)}$ are shared weight parameters, $\circ$ represents element-wise product of two matrices.

Remark 1 The units that described in this section is neither LSTM nor GRU since the input gate I_t or update gate Z_t introduces time duration d_t^{ij} as input, which is different from a simple constant bias in the first learnable bias in these gates. It is worth mentioning that adding bias to the gates are helpful to improve learning performance in LSTM [Jozefowicz et al., 2015], APM also use the trick as shown in F_t of Equation 4.

Remark 2 The term $\frac{d_t^{ij}}{d_t^{ij}+1}$ is a squashing mechanism, it normalizes d_t^{ij} from $(0,\infty)$ to $(0,1)$.

3.3.4 Ending Mark Interpretation

In context decoder, APM mentioned an ending mark "<EOA>" that indicates the termination decoding process. However, the ending mark is different from other marks, since in practice, "<EOA>" is represented in different symbols of behavior-based categorical clickstream, which as a label to involve classification of user actions.

Assume action paths are labeled by one-hot encoded ending marks $\mathrm{EOA}_1, \mathrm{EOA}_2, ..., \mathrm{EOA}_m$ and the last output of decoder hidden state is h_n, thus:

$$
\begin{aligned}
\hat{y} =& \operatorname{argmax}(\operatorname{softmax}(W^{(M)}h_n)) \\
\hat{y} \in& \{\mathrm{EOA}_1, \mathrm{EOA}_2, ..., \mathrm{EOA}_m\}
\end{aligned}
\tag{6}
$$

where $W^{(M)}$ is a weight parameter, and m is the number of ending mark categories.

3.4 Action Path Optimization

In traditional classification models, the argmax are used to select labels with the highest probability, scilicet, argmax selects predicted URLs with the highest probability of user action from decoder outputs. However, this method is under the condition of all outputs are independent in probability, which is not suitable for the action path optimization scenario.

In previous sections, APM feeds an input clickstream $(U_1^{ij}, U_2^{ij}, ..., U_t^{ij})$, and produce an output $(o_1, o_2, ..., o_m)$ that expect close to actual clickstream $(U_{t+1}^{ij}, U_{t+2}^{ij}, ..., U_n^{ij})$. Then the probability of expected clickstream is a conditional probability under the input clickstream. In other words, APM needs to solve an optimization problem:

$$
\begin{aligned}
& \operatorname*{argmax}_o p(o_1, o_2, ..., o_m | U_1^{ij}, U_2^{ij}, ..., U_t^{ij}) \\
=& \operatorname*{argmax}_o \prod_{k=1}^{m} p(o_k | U_1^{ij}, ..., U_t^{ij}, o_1, ..., o_{k-1}) \\
=& \operatorname*{argmax}_o \sum_{k=1}^{m} \log p(o_k | U_1^{ij}, ..., U_t^{ij}, o_1, ..., o_{k-1})
\end{aligned}
\tag{7}
$$

A heuristic approach can solve the optimization problem efficiently, namely beam search [Graves, 2012]. In each step of decoder output, **the approach reserves the top-k best combinations of URLs and eliminate the rest of URLs from evaluation, and finally selects k best clickstreams.** The pseudocode is given that adapts vanilla beam search to URL prediction search in Algorithm 1.

Algorithm 1: Output Clickstream Search

```
input  : Decoder outputs (o_1, o_2, ..., o_m),
         Number of candidates k
output: k clickstream candidates with highest probability
begin
    Initialize empty clickstreams list
    for o ∈ (o_1, o_2, ..., o_m) do
        Initialize empty candidates list
        for clickstream ∈ clickstreams do
            for page ∈ o do
                clickstream.append(page)
                p(clickstream) = log(p(clickstream)) + log(p(page))
                candidates.append([clickstream, p(clickstream)])
            end
        end
        ordered = sort(candidates, score in descending order)
        clickstreams = ordered[:k]
    end
end
```

Remark The algorithm produces a heuristic output with given clickstream context. The p(clickstream) was initialized as zero, and p(page) is produced by APM outputs. Combining with the *url2vec* embeddings, the prediction can heuristically optimize the click path of a specific user since the embeddings are trained over all possible action path. For instance, a distraction advertisement page will not appear after optimization because the embedding of the advertisement page is far from the desired page if embeddings are learned correctly.

As a reminder, readers should aware that the APM is only similar but different from sequence to sequence model, the major differences are:

1. URL embeddings are self-trained because they are not in a context of nature language. A pre-trained language model cannot be used based on transfer learning approaches. The embedding training initialize URL vectors one-hot encoded but get updated into a dense space before feeding into recurrent unit.

2. The hidden layer (recurrent unit) is not a standard LSTM or GRU because it also feeds scalar value of "time duration" into the network. This is a special processing of combining vectors and scalar inputs into APM, which makes the structure is different than LSTM or GRU.

3. The network is required to produce a behavior category token at the end of decoding, which needs categorial cross entropy.

4 Experiment

We must know. We will know.

David Hilbert

In this chapter, the author of the thesis rationalize the process of the lab study based on the theory of human information behavior. Next, the purpose of context-given web browsing tasks is construed for the subjects.

The lab study took place during the last two weeks of November, from 14/11/2018 to 29/11/2018 in Frauenlobstrasse 7a, a faculty building of Ludwig-Maximillians-Universitaet Muenchen. The action path data was collected by a self-developed embedded collector plugin installed in the mainstream browser, such as Google Chrome, on a self-provided desktop computer and a laptop.

In the lab study, the thesis selects three mainstream websites: Amazon, Medium, and Dribbble. These websites that cover categories for shopping, media consuming, and design brainstorming with design reasons (discussed later in Section 4.3). Then, the author of this thesis manually designed 35 reasonable tasks and finally selected nine context-given browsing tasks (three for each website, discussed in Section 4.3) to simulate three different kinds of proposed browsing behavior, namely *goal-oriented, fuzzy and exploring behaviors*. They are defined and discussed later in 4.2.

Each task requires participants to start from a starting page of a given website. The tasks do not restrict participants to using the given website only; they also allow participants to access websites outside the domain of the landing page to help they complete the task (this information is provided to participants before participation). Participants start browsing after they completely understand the requirements of each task. No interruption or question answered during the task. If the time limit of a task exceeded, subjects can either acquire more time to accomplish the task or give up if they feel that the task is too difficult.

The study is designed as a **within-subject** study. Thus every participant performs all tasks. To eliminate the learning effect due the extensive duration of using same websites, a Latin square [Cochran and Cox, 1950] is used in the thesis. for the devices (desktop and laptop) and tasks participation order for the subjects.

The lab study focused on 21 participants with a mean age of 23.04 (standard deviation of 3.216, min=18, and max=29). Of the participants, 10 were male and 11 were female. They were recruited anonymously and randomly selected via a mailing list.

4.1 Environment

The lab study used two self-provided devices: a desktop computer and a laptop. The reason for choosing two devices is that the study requires recording a complete clickstream during the study.

A major issue of mobile devices is that the operating system does not authorize the permission of allowance to collect data precisely over pages or user actions. Although Android devices can overpass system permission to privilege, the user behavior between iOS and Android devices has different personalities [Sandoiu, Ana, 2018]. Subjects exhibit [Reinfelder et al., 2014] abnormal awareness behavior regarding security and privacy issues when handling a newly provided Android device after they switch from an iOS device. Therefore, to eliminate this awareness, the study focus on desktop devices, which allows us to collects the clickstream data from browsers with plugin supports.

All modern browsers support plugin development, Google Chrome [StatCounter, 2018] has 61.7% market of market share of desktop browsers, while Apple Safari only possesses 15.0% of the market. Google Chrome is therefore dominant the desktop web browser market.

Hence, the author of the thesis decided to use Chrome to establish a plugin for collecting data. The questionnaire after the lab study indicates the subjects' browser usage share, as listed in Table 4.1. The result further support the browser selection.

Table 4.1: Browser usage shares of lab study subjects

	Google Chrome	Apple Safari	Mozilla Firefox	Microsoft Edge
Number	11	5	3	2
Percentage	52.38%	23.81%	14.29%	9.52%

4.2 Browsing Behaviors

Before explaining the design justification for the context-given browsing task, this section presents and discusses three types of user browsing behavior: **goal-oriented**, **exploring** and **fuzzy**.

These three terminologies are aggregated and incorporated from behaviors that have been determined in former qualitative research on web browsing behavior. These terminologies are based on the fundamental theory of interdisciplinary perspective information seeking behavior [Wilson, 1997], which was discussed in Section 2.3. Table 4.2 compares the terminology differences between former research and this thesis.

Table 4.2: Terminology comparison of information behavior on the web

Author	Terminologies	Terminologies	Terminologies	Main Factors
[Choo et al., 1999]	Formal search	Conditioned viewing; Informal search	Undirected viewing	Psychological; demographic; role-related environmental; source characteristics
[Johnson, Ross, 2017]	Directed browsing; Known-item search	Semi-directed browsing; Explorative seeking; "You do not know what you need"; Re-finding	Undirected Browsing	Behavior
This thesis	**Goal-oriented**	**Fuzzy**	**Exploring**	**Purpose**

To justify the terminologies, the follows combines the six qualitative activities from Ellis' Model [Ellis, 1989] and "information use" from Wilson's framework [Wilson, 1997] of information behavior theory to represent the summarized browsing behaviors:

Goal-oriented behavior *occurs when a user initiates a visiting session on the web caused by a determined objective in a specific context*, such as business work, social communication, university study, literature research, and so on.

Goal-oriented behavior indicates a piece of active information behavior. Instead of *formal search*, that only covers the phase of "monitoring" and "extracting" (or *directed browsing* and *known-item search* that covers "browsing" and "differentiating" or "monitoring" and "extracting" respectively), goal-oriented browsing behavior contains the entire life cycle of human information behavior starts from "starting" phase. By observing a browsing behavior, a determined "information use" can be observed and concluded.

For instance, a college student intentionally need a latest lecture slide (*information use* observed), the student then opens web browser, access college website (*starting*) and navigates to the lecture homepage (*chaining*, *browsing*, and *differentiating*). Finally, the student exit browsing after download the slides (*monitoring* and *extracting*).

Exploring behavior *occurs when a user initiates browsing session aimlessly with no clear observed extracting or information use during the session, the person greedily or breadth-first consumes and the content on the Web without any information extracting and information use*, such as media consuming, learning before using and so on.

Exploring browsing behavior indicates opposite behavior from goal-oriented browsing behavior. A more formal description of exploring behavior using Ellis' model, would note that the behavior represents "chaining" and "browsing" without "differentiating" and "extracting" from "starting" while information seeking.

For instance, a person who accesses an unknown utility web application (*starting*), may explore the functions one by one as well as what they can do while using the application (*chaining* and *browsing*).

Fuzzy behavior *occurs when a user initiates a visiting session for information use with non-systematic and incomplete prior knowledge that may involve ongoing browsing ongoing to update the framework of knowledge until final acquisition or abandon.*

Fuzzy behavior in browsing behavior is in between goal-oriented and exploring behaviors. Instead of only "chaining" and "browsing" from "starting", fuzzy behavior also engages "differentiating" or "monitoring" while information seeking.

For instance, a researcher may have heard a new technique proposed in another scientific field that may influence their research. That person may then opens a search engine (*starting* and *chaining*) to seek (*browsing*) existing (*differentiating*) follow-up research (*monitoring*). The browsing may end without information use because the technique is irrelevant to their research.

Remark Table 4.3 illustrates the existence of activities of three forms of browsing behavior. Note that "information needs" is not suggested in Wilson's theory [Wilson, 1981] because they can not be clearly observed before information seeking but sometimes may be observed after information use. Therefore information need is not considered in these terminologies.

Table 4.3: Existence of activities from Ellis' Model and information use in goal-oriented, exploring and fuzzy browsing behavior. The "exist", in the table, represents the existence of which activities contributes in which pattern. Information need is "N/A" because Wilson's theory does not suggest using information need to define terminology because it cannot be observed before information use.

Behaviors	Information Need	Information Seeking						Information Use
		Starting	Chaining	Browsing	Differentiating	Monitoring	Extracting	
Goal-oriented	N/A	Exist	Exist	Exist	Exist	Exist	Exist	Exist
Fuzzy	N/A	Exist	Exist	Exist	Exist	Exist		
Exploring	N/A	Exist	Exist	Exist				

4.3 Tasks Design

The author of the thesis designed 35 browsing tasks after conducting a pilot study. Nine tasks were selected for three websites: Amazon.com, Medium.com, and Dribbble.com because of the following reasons:

1. These three websites all have tasks that correspond to the three types of browsing behavior;

2. Each of the tasks can be finished in around 5 to 10 minutes according to the measurement of pilot study;

3. All these websites are mainstream websites that do not require significant professional domain knowledge to use.

In addition, the unselected tasks are listed in Appendix B.3.

4.3.1 Tasks of Goal-oriented Behavior

The author designed and selected an appropriate goal-oriented task for selected websites. Each task is designed with three designed information needs as the justification for information use.

Amazon.com *Assume your smartphone was broken and you have 1,200 euros as your budget. You want to buy an iPhone, a protection case, and a wireless charging dock. Look for these items and add them to your cart.*

This task initiates from the homepage of Amazon (*starting* and *chaining*), and contains three determined objective since a subject is required to add three specific items to the cart (*information use*). There are a few hidden considerations behind the task (*browsing* and *differentiating*), which makes the task more realistic (*monitoring* and *extracting*): a) There is a budget for this task, which requires subjects to consider the price of items instead of simply adding the first recommended item to cart. b) the starting page is amazon.com instead of amazon.de. This decision requires subjects to consider the exchange rate between U.S. dollars and euros for budgeting. c) There are some items cannot be shipped to Germany (the study took place in Germany). Subjects cannot add these items to the cart and should find other alternatives.

Medium.com *Assume you are making plans for your summer vacation. You want to visit Tokyo, Kyoto, and Osaka. You want to find out what kind of experience other people have had when traveling to these three places in Japan. Your task is to find three posts on traveling tips regarding these cities. Elevate a post if it is one of your choices.*

This task contains three determined purposes because there are three fixed traveling destination (*extracting* and *information use*). The task also involves a few considerations that increase the required interaction between the task to subjects: a) The website only offers an English version. Some Japanese characters may appear in an article. Thus, a translation website may be used during the study (*starting* and *chaining*). b) An article may contain numerous nouns, such as toponyms. Search engines may used during the study (*browsing*). c) Articles, that require a membership to access, cannot be elevated (*differentiating*).

Dribbble.com *You are hired at a cloud computing startup company. You receive assignment to design the logo of the company. Search for existing logos for inspiration and download three candidate logos that you like the most.*

The task also has three determined purpose because subjects are required to download three candidate trademarks (*extracting* and *information use*). During the participation, subjects must take a few implicit facts in to account: a) Subjects who unfamiliar with the term “Cloud Computing” must visit other explainations to determine the vision and mission of this type of company (*starting*). Subjects who are already familiar with the term still need to compare the designs made by other competitors (*chaining*, *browsing* and *differentiating*). b) Subjects should aware that some of the designs shared on the website are not suitable for trademark or icon design (*monitoring*).

4.3.2 Tasks of Exploring Behavior

Exploring tasks simply do not provide any deterministic objective, and all websites have a exploring task that is designed for subjects.

Amazon.com *Look for a product category that you are interested in and start browsing. Add three items that you would like to buy to your cart.*

Although the task do not require any specific items from the subjects, the task remains to have three different purposes because participants must add three items to the cart. This task is aimless because: all the tasks are not specifically informed to participants. They either do not have the needs to buy items or had needs of buy a specific category but do not have a product candicate yet. In any case, the description of the task request participants to start from a product category (*starting* and *chaining*), which avoids goal-oriented buying of a specific product.

Medium.com *Visit a category you are interested in and elevate three posts that you like.*

This task has a similar reason to the one as discussed in Amazon.com's exploring task (*starting* and *chaining*). Medium is a media website. Hence, visiting a specific article that read before participation is relatively difficult because all the content that is showed to users is updated daily. Thus, this task can be considered to be an exploring task.

Dribbble.com *Explore Dribbble and download the three images you like the most while you browse.*

Dribbble illustrates designs by using the image gallery (*starting* and *chaining*). The major difference between Dribbble and Google Image Search is that Dribbble is a user-centered content aggregation website, while Google Image Search is a simple content aggregation engine. Hence, there will be two different interactions in Dribbble: exploring designs based on keywords and categories or exploring designs based on users. The latter helps its user to finding similar design style. The task is aimless because the task simply describes nothing and lets participants explore their preferences for design styles.

4.3.3 Tasks of Fuzzy Behavior

Each of the selected websites also has a fuzzy task, and there are three major goals for each task that act as control conditions to subjects' action paths in the experiment.

Amazon.com *You want to buy a gift for your best friend as a birthday present. Add three items to your cart as candidate.*

The clarity of the task is stronger than the exploring task but is weaker than the goal-oriented task, because the task restricts participants from adding items for a specific purpose (birthday present) but does not point to any specific product (no *extracting*).

Medium.com *Assume you have an occasion to visit China for business. You are free to travel to China for a week and want to make a travel plan for that time frame.. Your task is to determine what kind of experiences other people have had when visiting to secondary cities or towns in China, then decide on three cities you want to visit (excluding Beijing, Shanghai, Guangzhou, and Shenzhen). Elevate a post if it helped you to decide.*

The clearness of the task is stronger than exploring the task because it asks a participant to explore a non-deterministic direction of looking for secondary cities (no *extracting*). However, the clearness of the task is weaker than the goal-oriented task because the secondary cities described in Medium's user posts are unclear, and participants are supposed to make decisions themselves. Furthermore, this task pertains to traveling around China for a week. Cities cannot be randomly selected because making travel plans requires consideration of a city's geographic location.

Dribbble.com *You are preparing a presentation and need one picture for each of these animals: cat, dog, and ant. Download the three pictures you like the most.*

The task has three purposes of downloading images of animals, which restricts participant to a specific direction. Thus, the clearness of the task is stronger than the exploring task. However, the task describes a scenario of using these images in a presentation. Hence participants must consider the continuity of the design style, which makes the clearness of the task weaker than the goal-oriented task (no *extracting*).

5 Evaluation

> If a machine is expected to be infallible, it cannot also be intelligent.
>
> Alan Turing

In this chapter, the author conduct evaluations of the collected data. The data is collected from 21 subjects, and 189 clickstream data are collected in total. Each clickstream contains action-level data with a stay duration of a specific page, for instance, the study also collected a URL as a step of clickstream if a participant uses back button rollback to a previously visited page without requesting server. *A clickstream also has a subjective difficulty score that measured from the questionnaire through a self-rating scale (shown in Appendix B) after the completion of each task.*

5.1 Subjective Task Difficulty

This section discusses the subjective task difficulty qualitatively and quantitatively. Figure 5.1 illustrates a normalized (raw scores are listed in Appendix C Table C.1) subjective difficulty score with respect to all tasks.

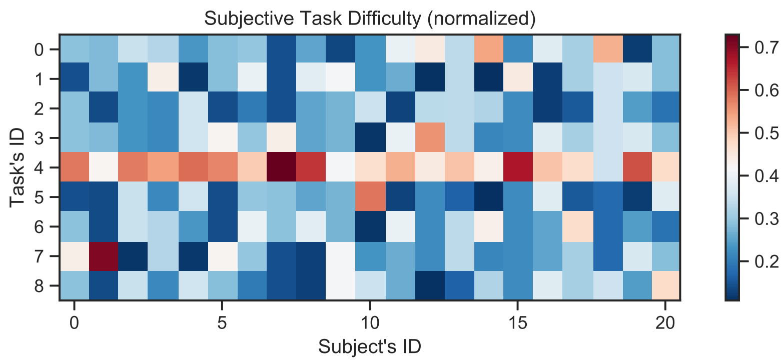

Figure 5.1: Subjective difficulty score: each column indicates an individual subject and each row indicates a browsing task. *TasksID* from 0 to 8 represent Amazon Goal Oriented Task, Amazon Fuzzy Task, Amazon Exploring Task; Medium Goal Oriented Task, Medium Fuzzy Task, Medium Exploring Task, Dribbble Goal Oriented Task, Dribbble Fuzzy Task, and Dribbble Exploring Task respectively. From this heat map, one can observe Medium Fuzzy Task is the most challenging task according to the subjects voted subjective difficulty, a Mann-Whitney U significant test justifies this observation.

The purpose of measuring task difficulty is to give a verification of task design, and understanding how subjects votes the difficulty in different browsing behaviors. Therefore, a significant test is considered with the null hypothesis (H_0): the difficulty of fuzzy task is not significantly greater than exploring task and alternative hypothesis (H_1): the difficulty of fuzzy task is significantly greater than exploring task.

A non-parametric one-tailed Mann-Whitney U test [Mann and Whitney, 1947] is conducted as follows. Under the null hypothesis, $p = 2.54 \times 10^{-5} < 0.05$, reject H_0. Similarly, one can compare difficulty score on goal oriented task and exploring task (with corresponding hypothesis, $p = 0.00534 < 0.05$), difficulty score on fuzzy task and goal oriented task (with corresponding hypothesis, $p = 0.0145 < 0.05$), all rejects H_0. Therefore one can conclude that the task difficulty is ordered as follows: *difficulty of fuzzy task > difficulty of goal oriented task > difficulty of* *exploring task*, which means exploring tasks have lower effort in clickstream, and effort of doing fuzzy task gains highest effort.

5.2 Browsing Behavior Classification

As discussed in Section 4.3, three types of browsing behavior are described. In this section, the author of the thesis provides two types of evaluations to interpret the browsing behavior classification.

First, the thesis evaluates the indication of general features browsing behavior: task efficiency (Section 3.1), number of actions in an action path as well as the total stay duration in the action path. Then APM is implemented by using the action-level clickstream data and stay duration of each page, which was described in Section 3.3.3 and 3.3.4.

5.2.1 Interpretation based on General Features

As a baseline and a comparison to the APM for the classification performance, the *completion efficiency*, *total time duration of a task* as well as *total number of actions of a task* are used for browsing behavior classification in support vector machine (SVM) [Suykens and Vandewalle, 1999].

Note that the shortest path of entire clickstream defines the completion efficiency, moreover, the completion efficiency can only be determined if and only if the clickstream is given, in a sense, it carries latent information of browsing behavior.

SVM with the polynomial kernel is applied with gird-search. The best classification precision is 0.53 (with $C = 4.5$ and $\gamma = 1.5$, which are the searched hyper-parameters in SVM model). The micro average F1 score is also 0.53, which is better than random (0.33). The t-SNE visualization is showed with pairs of features for graphical insights in Figure 5.2.

To better understand the meaning of classification, a randomized decision tree is also applied (ExtraTreesClassifier in *scikit-learn* [Pedregosa et al., 2011] with default parameter settings) that gives the importance of the used features: ***total time duration and number of actions of a task are more important than self-defined completion efficiency.***

Moreover, one-tailed Mann-Whitney U test is used in each pair of these features. For instance, the null hypothesis (H_0): the completion efficiency of the goal-oriented task is not significantly greater than exploring task. The result shows $p = 0.0019 < 0.05$ reject H_0, which means the completion efficiency of goal-oriented task is significant efficient than than exploring task.

Similarly, one can conduct the significant test with similar hypothesis to all comparable combinations as listed in Table 5.1, 5.2, and 5.3.

Table 5.1: One-tailed significant tests for completion efficiency in different browsing behaviors. The null hypothesis in this table, for instance, completion efficiency of the fuzzy task is *not* significant efficient than the goal-oriented task, the result $p = 0.45 > 0.05$ which means accept $H0$. Similar to others.

v.s.	efficiency goal	efficiency fuzzy	efficiency explore
efficiency goal	N/A	reject	reject
efficiency fuzzy	accept	N/A	reject
efficiency explore	accept	accept	N/A

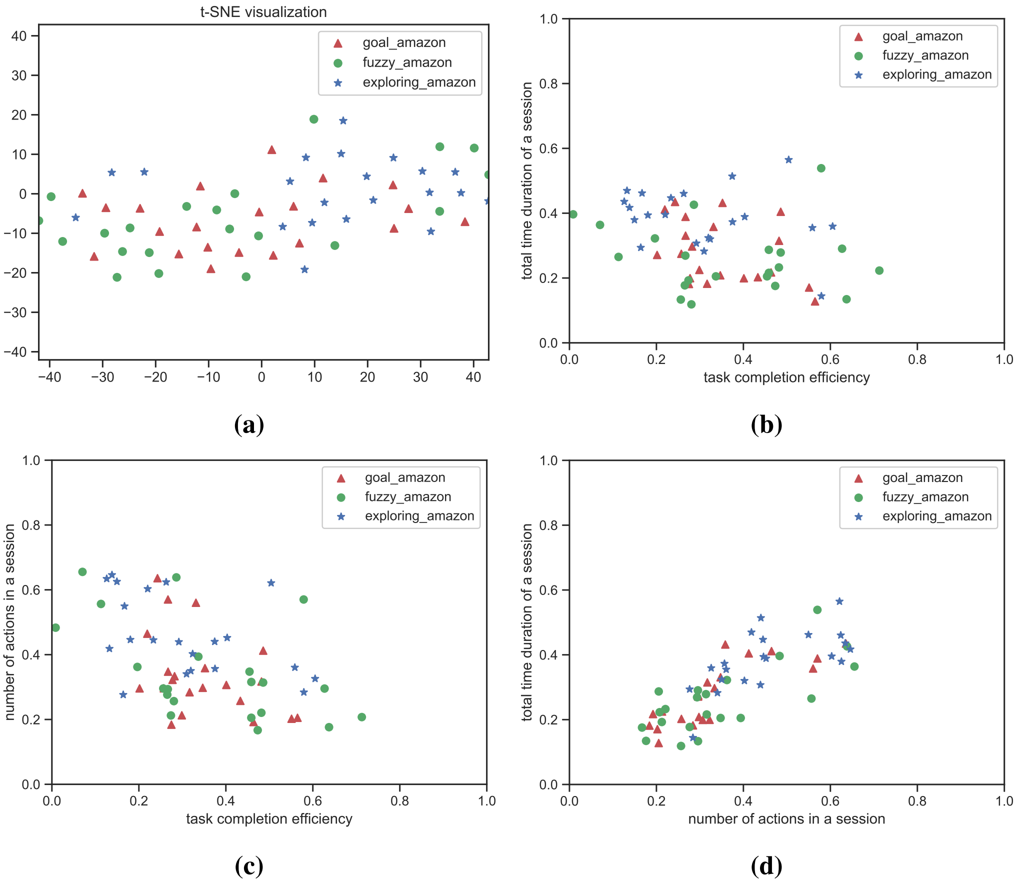

Figure 5.2: In these figures, 5.2a shows the t-SNE projection of completion efficiency, total time duration and number of actions for three different behavior; 5.2b is a 2D comparison of using completion efficiency and total time duration; 5.2c provides a 2D comparison of using completion efficiency and number of actions; 5.2d shows a 2D comparison of using number of actions and total time duration. From t-SNE visualization, one can observe that exploring tasks tend to centralized on the right and goal-oriented tasks and fuzzy tasks tend to centralized on the left, which indicates that exploring behaviors tend to classifiable comparing to the other two behaviors. According to the rest of feature comparison visualizations, the completion efficiency and total time duration contributes more to interpret exploring behavior, and the number of actions tent to contributes more to interpret goal-oriented task.

Table 5.2: One-tailed significant tests for total stay duration of a task in different browsing behaviors. The null hypothesis in this table, for instance, total stay duration of the fuzzy task is *not* significant stay longer than goal-oriented task, the result $p = 0.41 > 0.05$ which means accept $H0$. Similar to others.

v.s.	duration goal	duration fuzzy	duration explore
duration goal	N/A	reject	reject
duration fuzzy	accept	N/A	reject
duration explore	accept	accept	N/A

Table 5.3: One-tailed significant tests for total number of actions of a task in different browsing behaviors. The null hypothesis in this table, for instance, total number of actions of the fuzzy task is *not* significant performs more actions than the goal-oriented task, the result $p = 0.019 < 0.05$ which means reject $H0$. Similar to others.

v.s.	actions goal	actions fuzzy	actions explore
actions goal	N/A	accept	reject
actions fuzzy	reject	N/A	accept
actions explore	accept	reject	N/A

As a summary, the insights of each feature are concluded as follows:

- **Completion efficiency**: the completion efficiency of goal-oriented and fuzzy behavior is significant efficient than exploring behavior;
- **Number of actions**: the number of actions of goal-oriented behavior is significantly lower than fuzzy and exploring behaviors.
- **Total stay duration**: the total stay duration of exploring behavior is significantly higher than goal-oriented and fuzzy behaviors.

Furthermore, the completion efficiency and total stay duration are more critical than others for indication of exploring behavior, also, the number of actions are more important than others for indication of goal-oriented behavior.

5.2.2 Interpretation based on Action Path

To use the full capacity of action path data and learn the deep inside structures of an action path, this section uses the entire clickstream and its corresponding action-level stay duration as input, three ending mark (<EOA_GOAL>, <EOA_FUZZY>, and <EOA_EXPLORE>) as classification outputs. Then a single GRU-like layer based APM is presented for the classification of the three types of browsing behaviors.

The tunable training parameters are equivalent to standard GRU: The latent dimension is 10, the training process feeds 132 clickstreams as training data, 38 clickstreams as validation, then propagate 500 epochs with a batch size of 32. In the training process, Adam optimizer is used, categorical cross-entropy loss as well as L2 regularizer (with 0.0000001) with early stopping (patient 1000), the total number of trainable parameters is 90323.

At the end of training, 19 clickstreams as the testing dataset are evaluated. **The APM archived precision of 1.00%** of browsing behaviors classification, i.e., 100% of accurate. Note that the training set is randomly selected from all participants, which is supervised with k-Fold cross validation [Kohavi, 1995] while training, the reason for using k-Fold other than else is discussed in Section 7.2.

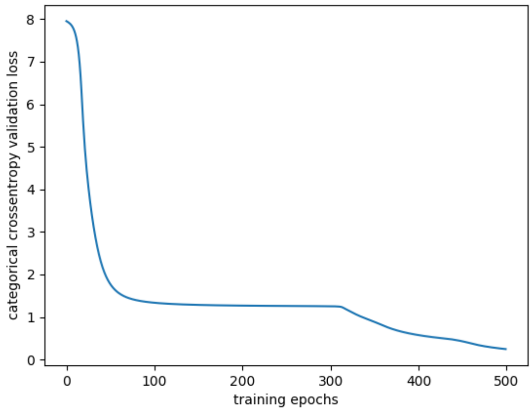

Figure 5.3: Categorical cross-entropy validation loss curve while 500 epochs. The curves indicates the training process is not an overfitting since the loss is not increasing but keep reducing.

One can observe that the training process is not an overfit, and the validation loss is still not increase after 500 epochs. Thus, a single GRU-like layer in APM remains a large expressive generalization performance (100% accurate for three browsing behavior classification). Therefore the author expect to collect more data to verify whether the APM applicable to a large dataset.

In addition, the APM feeds the entire clickstream and time duration as inputs, therefore the entire clickstream contains pieces of information regarding the number of actions as well as completion efficiency and more latent pieces of information. Consequently, one can conclude that the APM works *perfectly on the classification of three different browsing behavior*. Since the experiment is only designed for three types of behavior, and the learning curve shows the APM still has the capacity and generalization ability to classify more precise categories of browsing behavior, a future investigation on more categories may be worthwhile.

5.3 Optimal Action Path Context

This section evaluates the APM with limited action path context, where the feeding action path is limited based on a split ratio. For instance, if a split ratio is 0.8, then 80% of an action path is fed into the APM, then predict the rest of 20% actions. Figure 5.4 illustrates the best accuracy archived from a single layer APM when used with a different split ratio.

This figure illustrates, with more context of clickstream feeds into the APM, the model receives more pieces of information of the clickstream. Therefore much higher accuracy the APM can archive for prediction. The accuracy of the APM evaluated here is **a greedy search accuracy**, which performs an element-wise comparison between predicted clickstream and ground truth clickstream, and the accuracy is the number of corrected predictions divided by the total number of prediction steps. Note that the greedy search is used in this evaluation because the accuracy is compares ground truth and behaviors that APM learnt, nevertheless, the beam search of APM that proposed in Section 3.4 is used for optimizing user actions.

An accuracy that higher than 25% is acceptable in the prediction task since it indicates a quarter of future movements are predicted correctly. On the right side of the figure, the APM archived >60% accuracy of 3 to 5 future steps prediction.

Note that the prediction is still not overfitting to the dataset. Figure 5.5 illustrates the loss curve while training over 1500 epochs with three steps of prediction (split ratio 0.97). The loss starts to increase after almost 200 epochs, which may be represented to overfitting. Nevertheless, one can observe that the loss decreases down to a similar level of early training. It archived a better performance (almost 60.0% of precision) than previous, which indicates the training process may

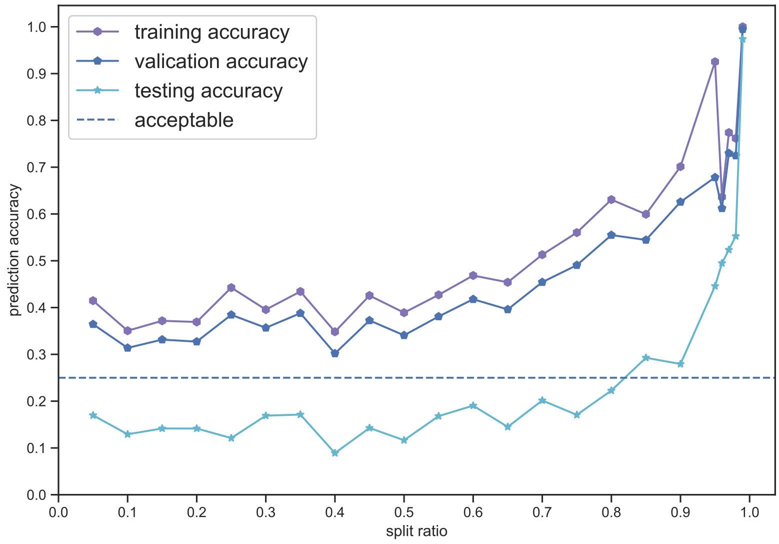

Figure 5.4: Prediction accuracy with a limited context of input. This figure illustrates, with more context of clickstream known to the APM, more information to the model, and therefore much higher accuracy can be archived. The accuracy that evaluated here is a greedy search accuracy, and thus higher than 25% of prediction accuracy is baseline, i.e., a quarter of future movements are predicted correctly. On the right side of the figure, the APM archived >60% accuracy of 3 to 5 future steps prediction. Classification is a particular case in this figure where the split ratio is equal to 0.99.

re-parameterize the APM while training and archive better performance for predictions.

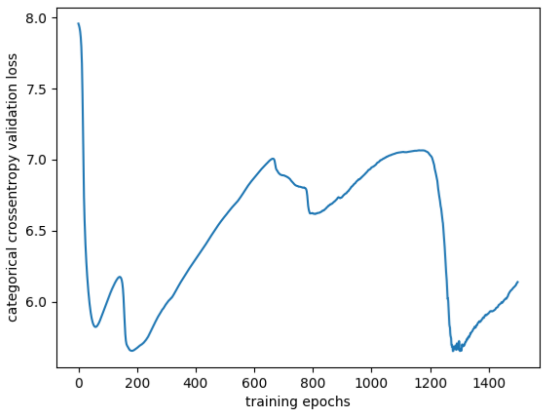

Figure 5.5: *Validation loss* curve when the split ratio is 0.97. The loss indicates the model may be re-parameterized while training and archive better performance for predictions.

5.4 Action Path Visualization

This section visualizes the actual action path of users and discusses the behavior qualitatively. In total, 189 clickstreams are collected, which is not possible to illustrate all of them in the thesis, the section selects the typical clickstreams to discuss and provides a visualization tool (see Appendix A) to help readers to explore all action paths.

5.4.1 Individual Common Patterns

Pattern of "cluster" The first pattern one can observe from the goal-oriented task clickstream is called "cluster". In Figure 5.6 and 5.7, the visualization shows different clustered intents in Amazon's goal-oriented task. Formally, *a pattern is called "cluster" if and only if it is a partition of an action path that is connected with rest of the action path through a single node.*

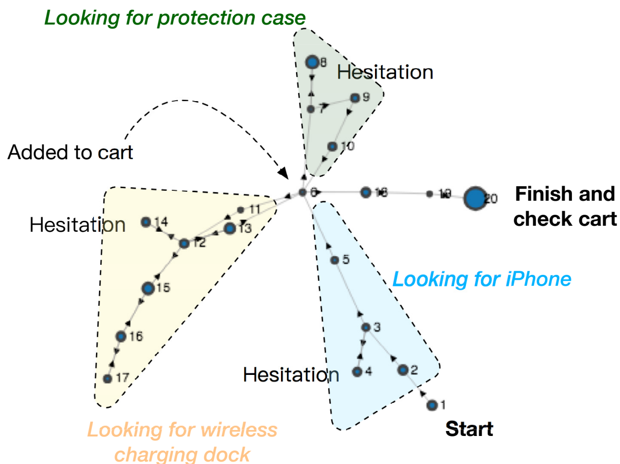

Figure 5.6: Patterns of cluster and hesitation of an action path. This figure visualizes an action path in goal-oriented Amazon's task. The visualized graph can be partitioned into four subgraphs and three of them are cluster pattern that is a representing of different shopping intent, which is exactly as same as the task design. Further, each cluster contains a hesitation pattern as labeled in the figure, for instance, the node labeled with 4, 8, 14 are hesitation. Besides, the number of a node is a representative of a chronological serial number of user actions.

One can easily discriminate the user browsing for different intent in a different cluster, and then finally went to the cart without backtracking.

Pattern of "hesitation" Beyond the cluster pattern, "hesitation" pattern is also observed in goal-oriented tasks where a short child path branch from its parent node in each intent cluster, e.g. node 4, 8, 14 in Figure 5.6 and node 5, 16 in Figure 5.7, which suggests "hesitation" is a pattern that more often appears in the goal-oriented task within a "cluster". Formally, *a pattern is called "hesitation" if and only if it is an acyclic list and not in a star that joint with a cluster or a ring and the number of its nodes is less than any of the existed cluster.*

Pattern of "ring" and "star" Similarly, in the fuzzy and exploring task, two common patterns "ring" and "star" patterns are observed that more often to appear in fuzzy and exploring tasks. Formally, *a pattern is called "ring" if and only if it is a list without connection to a cluster and starting node is not joint with ending node; a pattern is called "star" if and only if it is a spanning tree of an action path that a non-leaf node contains more than one child.*

Figure 5.8 illustrates an action path of Amazon's fuzzy task (purple nodes) and an action path of Dribbble's exploring task (orange nodes), both from same participants. One can observe "ring" and "star" patterns in the figure as highlighted through the gray area surrounded by a dashed line.

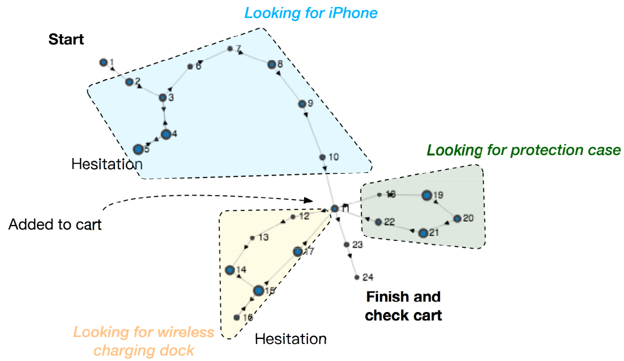

Figure 5.7: Patterns of cluster and hesitation of an action path. This figure visualizes an action path in goal-oriented Amazon's task. The visualized graph can be partitioned into four subgraphs and three of them are cluster pattern that is a representing of different shopping intent, which is exactly as same as the task design. Further, two of the clusters contain a hesitation pattern as labeled in the figure, for instance, the node labeled with 5, 16 is hesitation. Besides, the number of a node is a representative of a chronological serial number of user actions.

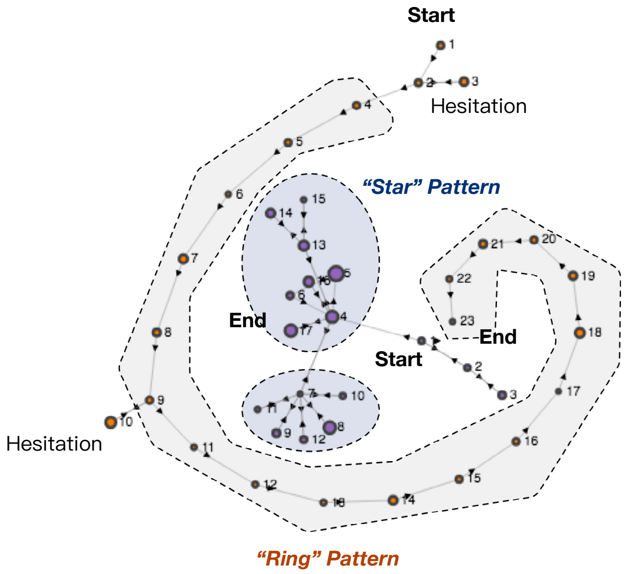

Figure 5.8: Patterns of ring and star of an action path. The figure visualizes an action path in Amazon's fuzzy browsing task (purple nodes) and Dribbble's exploring tasks (orange nodes). The visualized action path of exploring task is a linked list with few hesitations (node 3 and 10). The action path of fuzzy task contains two star patterns (roots are 4 and 7). As same as other visualizations, the number of a node is a representative of a chronological serial number of user actions.

Similarly, as one more illustration, Figure 5.9 gives action paths in the same tasks but from another participant that the purple nodes represent actions in Amazon's fuzzy task action path and orange nodes represent actions in Dribbble's exploring task action path.

In addition, even though the author observed that the number of star pattern is more often to appear in fuzzy tasks and ring pattern is more often to appear in exploring tasks. The author argues that this is because, in fuzzy tasks, participants can identify the information uses, therefore the star pattern is more often to appear since it produces many backtracking behavior and causes the "differentiating" activity. However, in the exploring task, there is no explicit information uses described the exploring task, therefore participants keep exploring deeper and deeper from the starting page without backtracking, the star pattern appears when a participant has multiple interests on different pages that referred from the same page.

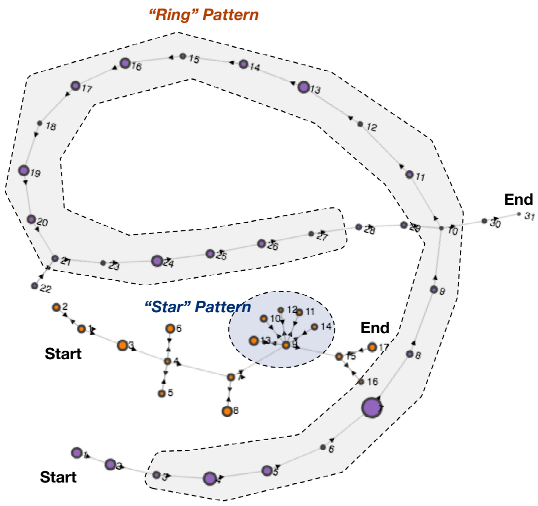

Figure 5.9: Patterns of ring and star of an action path. The figure visualizes an action path of a different participant in Amazon's fuzzy browsing task (purple nodes) and Dribbble's exploring tasks (orange nodes). The visualized action path of exploring task contains a star pattern where the root is 9. The action path of fuzzy task contains a cyclic ring pattern that with a single hesitation in node 22. As same as other visualizations, the number of a node is a representative of a chronological serial number of user actions.

In summary, one can conclude that:

1. Goal-oriented browsing behavior contains common patterns of "cluster", and each cluster tend to indicate a specific intent;
2. Fuzzy and exploring behavior two common pattern of "ring" and "star", however, ring pattern is more often to appear in exploring behavior and star pattern is more often to appear in fuzzy behavior;
3. Pattern of "hesitation" usually attached to a cluster or a ring but not appear in a star.

5.4.2 Cross user Overlap Patterns

In the previous discussion, the common patterns are discussed that appears in individuals. Nevertheless, it is still interesting to explore how action paths are manifest to multiple participants. Fortunately, there are intersections among multiple subjects.

Pattern of "overlap" occurs when observing action paths on multiple participants. Figure 5.10 and 5.11 are the action paths visualized for **the same four participants** in Medium's goal-oriented task, and Dribbble's exploring task respectively. One can define a $n-$overlap ratio is the number of blackening nodes divided by the total number of nodes in the action paths of n participants. According to the definition, the maximum number of $4-$overlap ratio is 100.00%, and the minimum $4-$overlap ratio is 0.00%.

However, the highest $4-$overlap ratio that observed from the dataset is 11.84% in the goal-oriented task. The lowest $4-$overlap ratio is 0.00% when compare two different tasks. Therefore, the author argue that the browsing behavior tends to be *user-specific* even users have the same goal in a task. However, they still share similar overlaps which suggest *common interests* between subjects.

In exploring task, the 4−highest overlap ratio is 1.15%, which is showed in Figure 5.11. The only common blacken node the starting page. This observation suggests that exploring browsing behavior is highly user-specific. Therefore, in conclusion, the overlap pattern of action path among multiple users suggests:

- Browsing behavior tends to be user-specific. However, it is unclear whether it is user-specific because the APM have an issue with lack of data, which is discussed in Section 7.3.

- Specifically, in goal-oriented browsing behavior, one can observe common interests between multiple subjects, whereas the exploring tasks have no intersection between subjects.

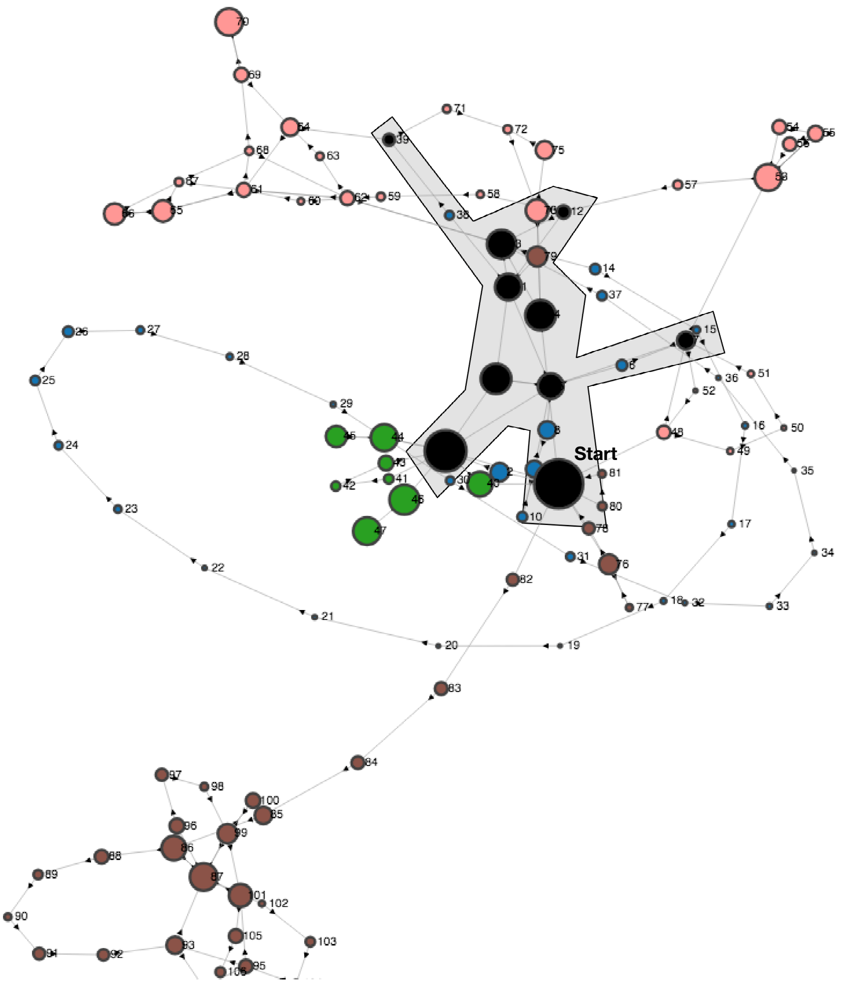

Figure 5.10: Example of "overlap" pattern in Medium's goal-oriented task: This figure visualizes the clickstream intersection of four participants at Medium's goal-oriented task. Each color represents an individual clickstream except black nodes, which represents the overlapping of different clickstreams. The overlap ratio of this graph is 9.43%.

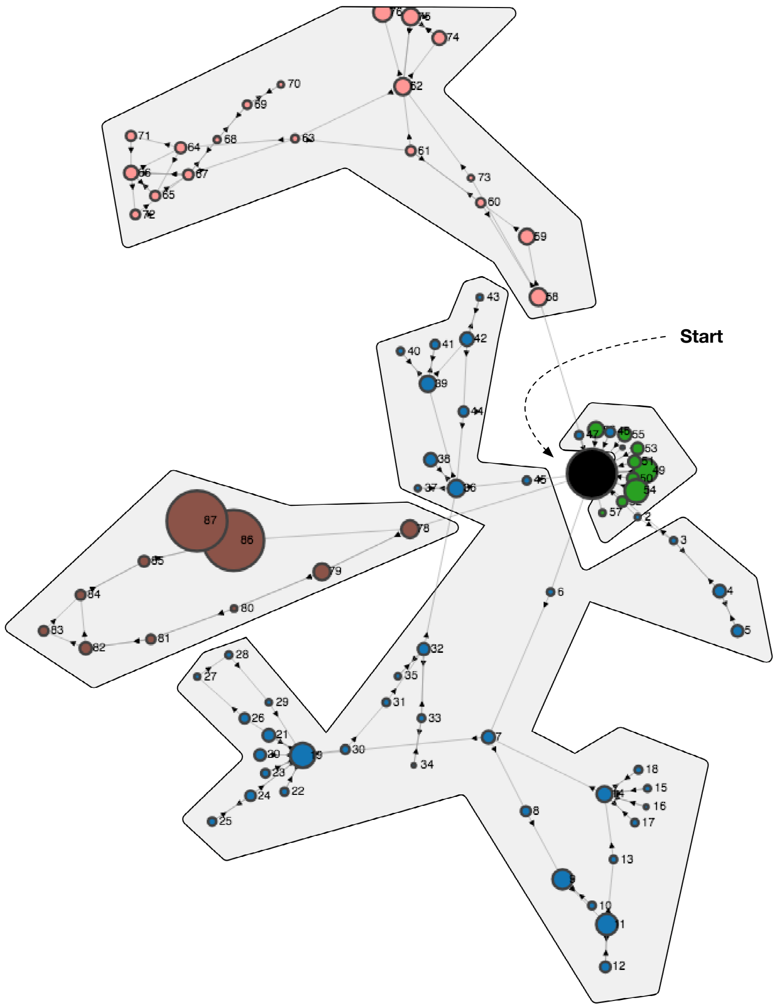

Figure 5.11: Example of “overlap” pattern in Dribbble’s exploring task: This figure visualizes the clickstream intersection of four participants at Dribbble’s exploring task. Each color represents an individual clickstream except blacken nodes, which represents the overlapping of different clickstreams. The overlap ratio of this graph is 1.15%.

Remark Table 5.4 shows an analysis of all observed patterns based on Ellis’ model, which explains why these patterns exist and how they contribute to the APM.

Table 5.4: Existence of activities from Ellis’ Model and information use in the observed patterns The “exist”, in the table, represents the existence of which activities contributes in which pattern. The “observed” indicates that the information need is observed from browsing behavior.

Behaviors	Information Need	Information Seeking						Information Use
		Starting	Chaining	Browsing	Differentiating	Monitoring	Extracting	
cluster	observed				Exist	Exist	Exist	Exist
star			Exist	Exist	Exist			
ring		Exist	Exist					
hesitation	observed		Exist		Exist	Exist		
overlap	observed						Exist	Exist

- For “cluster” pattern, as discussed before, information need can be observed from action path behavior, and the differentiating and monitoring contributes to the partitioning character of the pattern and extracting and information then contributes to the short ring and hesitations because the information is specified clearly.

- For “star” pattern, one can neither observe information need from action path nor did the participant uses information that finds in the star pattern. In Ellis’ model, chaining, browsing and differentiating contributes to this pattern since the depth from root to leaf node are small.
- For “ring” pattern, one can neither observe information need or information use. The user explores deeper and deeper along the ring until the user exit the browsing session.
- For “hesitation”, it connects to ring and cluster pattern, therefore they have common activities of chaining, differentiating and monitoring. However, information from hesitations are not used, but one can easily observe the hesitation.
- For “overlap”, one can observe common interests, which indicates information needs and use, the extracting and information use contributes more to represent this behavior.

Combining with Table 4.3, “cluster” pattern and “overlap” pattern essentially contributes to goal-oriented browsing behavior since they share common activities in this behavior, “star” and “ring” patterns contributes more on fuzzy and exploring tasks since their activities are more close to these browsing behaviors. Besides, as discussed before, these patterns cannot be observed with explicit information use. The “hesitation” pattern appears in “star”, “ring” and “cluster” pattern because they have common activities, such as “chaining” and “differentiating”.

6 Applications

Simplicity is complicated.

Rob Pike

In this chapter, the author first introduces a possible application of the proposed model. This includes the implemented features, how the model could benefit a user, as well as the architecture and data flow of the application. In the second part of this chapter, the author formalize and discuss the possibilities and benefits of being a standard web API for web developers and website designer.

6.1 Client-side Browser Plugin

This thesis developed a client-side browser plugin as a illustration of the model application. The plugin is an intelligent system that proactively serves its user and provides proactive notifications based on the historical actions in a session when browsing behavior is detected to goal-oriented or fuzzy behavior, as illustrated in Figure 6.1.

The user can either select "Yes" and navigate to the most likely page that they will visit in the future, or select "No" to ignore the notification and mark it as an invalid detection. The plugin serves the user only if the browsing behavior is clearly detected to forbear massive notification that disturb the user. The author argue that the plugin is only a supplement for improving browsing experience but is not always necessary. For instance in exploring behavior, a user's information need may not be clearly observed and the recommendations may not useful. One of the benefits of the plugin is to proactively help the user become efficient and reach the destination as fast as possible in the goal-oriented browsing.

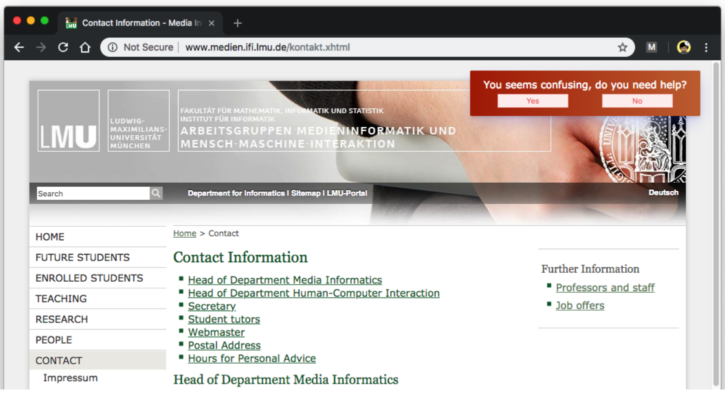

Figure 6.1: Proactive notification: The plugin injects monitor script when the page is loaded, and then serve user giving notification when detecting fuzzy browsing behavior.

In Figure 6.2, other than proactive notification, users can always open a popup page provided by the plugin. The popup page enables another interaction that privides the predicted needs based on historical user actions. A user can always interact with the plugin and retrieve the possible needs and browsing status in the current session. This information is helpful to the plugin user because a user can understand the current status of web browsing, which implicitly allows the person to better focus on whether the person is detected as a form of exploring browsing behavior.

The implementation and architecture is not simple although it provides a small feature that exhibits context and future information for the user. Figure 6.3 illustrates the implemented architecture of the plugin.

First of all, the plugin daemon process will inject monitoring script (*step2*) into the newly opened page (*step1*). When the user starts browsing and interacting (*step3*), the injected script will

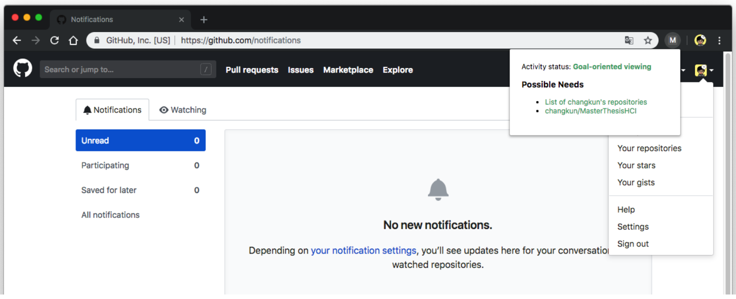

Figure 6.2: The plugin provided popup page: users can always open the page to understand the current status of browsing and predicted needs based on historical actions in a browsing session. In this case, the detected browsing behavior is under a goal-oriented browsing, and predicted actions are accessing the page of public GitHub repositories and accessing a specific repository.

report the referring of previous visited URL, current URL and stay duration to the daemon process of the plugin (*step4*).

Afterwards, the daemon process will report the referring information to the plugin server webhook (*step5*). Next, the webhook will immediately request the intra prediction microservice (*step6*) and result in a prediction (*step7*), which will then respond the prediction result to the daemon process with a pre-trained model (*step8*). Therefore the daemon process can decide if a proactive notification should be presented to the user or whether it should simply update its popup page for illustration (*step9*).

Since the prediction service received a new user action, it stores the action into a database subsequently for the model update (*step10*). Because of the cost of training a new model, the prediction service can decide to trigger the training service to retrain the model if it has already received enough new data (*step11*).

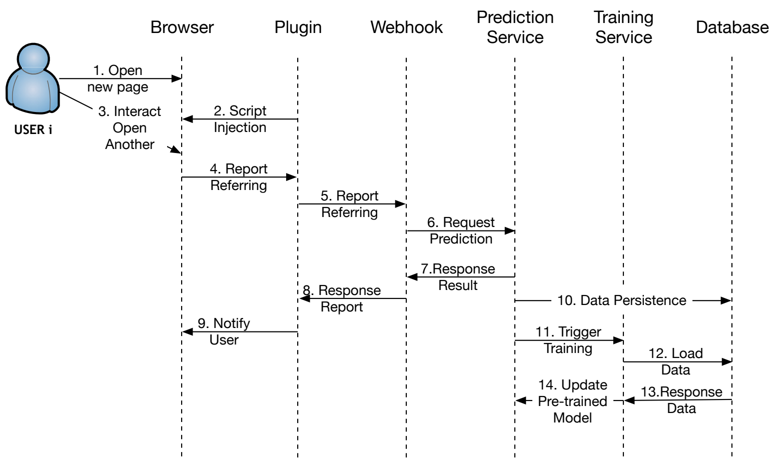

Figure 6.3: The implemented architecture of the plugin: This is the data flow illustration, Step 1 and 3 are user actions, and the rest of the steps are automatically triggered by each component of the plugin. For instance, the browser daemon process and plugin background process as two part of client-side components, the webhook, prediction service and training service are backend microservices running on a server.

Furthermore, the training service uses the pre-trained model as a base model to initiate the training by requesting newly created data from database (*step12* and *step13*), similar to the idea of transfer learning. After the training has achieved performance that is competitive to the pre-trained model, the training service will update the newly trained model to the prediction service (*step14*), which serves future prediction requests.

As one can observe from the architecture, the infrastructure is not as simple as the plugin feature intends to provide. Therefore, the author argue that the plugin feature is a feature that only browser manufacturers can provide. In the following section, the thesis formalize and discuss the possibilities of the plugin feature as a web API.

6.2 Web API Standardization and Platform-as-a-Service

Web API is a generic term used in various fields of development. Web API in the context of web browsers refer to the APIs provided by browser manufacturers to developers that helps with web applications. These APIs can even close help with manipulating hardware, for instance, WebAssembly [W3C, 2018].

Currently, there are experimental standard web APIs such as web speech APIs [Shires, Glen and Jaegenstedt, Philip, 2018] that integrate complex features to web developers and only have Google Chrome (after version 24) support. The specification proposal was initiated by Google. According to the source code of Chromium Kernel, the APIs are implemented based on the speech recognition service provided by Google Cloud Platform [1], which indicates that browser APIs do not only provide interfaces to the hardware but also access cloud platform services, i.e. Platform-as-a-Service integrated APIs.

The plugin that was illustrated in Section 6.1 can also be integrated as a PaaS API that is embedded into web browsers, which simplifies the infrastructure of the plugin. Developers can simply call the standardized API to report current user actions and obtain a response about current behavior status as well as the prediction of future movement or actions; see Figure 6.4 for the diagrams.

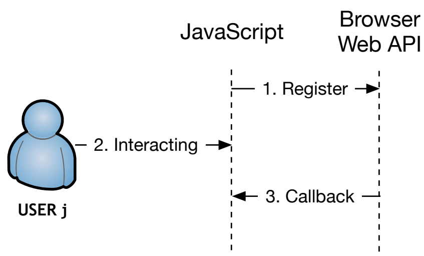

Figure 6.4: Usage overview of standardized BrowsingBehavior API

Defining the specification of the PaaS API aims enable web developers to use a web browser to monitor the future actions of their users. Developers can use the predicted actions to dynamically change the UI elements and improve the user experience of their product. the non-normative web API design of the browsing behavior predictor is discussed, which seeks to keep the API to a minimum.

6.2.1 The *BrowsingBehavior* Interface

The browsing behavior interface is a scripted web API for resulting in a monitored browsing session, which is presented in Code 1.

```
[Exposed=Window, Constructor]
interface BrowsingBehavior : EventTarget {

    // methods to drive browsing behavior response
```

[1] https://github.com/chromium/chromium/blob/83928864c18362a4b0f84bad9bee4104f4655430/content/browser/speech/speech_recognition_engine.cc#L35, last accessed on January 03, 2019

```
    void start();
    void stop();
    void pause();
    void resume();

    // event methods
    attribute EventHandler onBrowsingStart;
    attribute EventHandler onBrowsingEnd;
    attribute EventHandler onBrowsingPause;
    attribute EventHandler onBrowsingResume;
    attribute EventHandler onResult;
}
```

Code 1: BrowsingBehavior Interface

***start()* method** When the start method is called, it represents the moment in time the web application wishes to begin monitoring user's actions. Then every step when a user was making moves, the *EventHandler onResult* will produce a standard prediction and classification of user browsing behavior. Further, the *EventHandler onBrowsingStart* will be called immediately after calling this method and before resulting a prediction result, which gives a barrier in between of calling *start* and callback *onResult*.

***stop()* method** When the stop method is called, it represents the instruction to browsing behavior service to stop monitoring user actions, and resulting in a final prediction in the *EventHandler onBrowsingEnd*.

***pause()* method** This method is used to ignoring the upcoming user actions to pauses the monitoring of user actions, and resulting in a prediction in the *EventHandler onBrowsingPause*.

***resume()* method** This method resumes the paused *BrowsingBehavior* object and recovers the monitoring of user actions. Before monitoring is fully recovered, the *EventHandler onBrowsingResume* will be called.

The primary consideration of designing these four methods is to restrict abuse of the APIs. Similar to cookie, speech recognition APIs, a website should acquire an authorization from their user; otherwise, the API cannot monitor any user actions on the web, which partially solves the issue of privacy and security. more concerns about the feature are discussed in Chapter 7.

6.2.2 *onResult* callback

onResult callback passes the prediction after the browser user acted. The prediction result consists of two parts: behavior and future movements.

The *behavior* attribute of the result object is a JSON object that contains confidence level, i.e., classification probability, and a enumerate *category* attribute that indicate a finite set of user browsing behaviors, i.e., goal-oriented, fuzzy or exploring.

```
{
    "behavior": {
        "confidence": float64,
        "category": string,
    },
    "futures": [
        {
            "confidence": float64,
```

```
                "actions": array[string],
            },
            {
                "confidence": float64,
                "actions": array[string],
            },
            ...
        ]
}
```

Code 2: Result object of onResult callback

The *futures* attribute of the result object is an ordered JSON object that from the highest *confidence* to lowest confidence and the *confidence* is a floating number from minimum 0 to maximum 1. Meanwhile, the *actions* attribute in a JSON object of an item of *futures* array is an array of possible actions of URLs that ordered in chronologic order, the first element represents the next immediate action, and the last element represents the final action in the session, as shown in Code 2.

```
{
    "device_id": string,
    "previous_url": string,
    "current_url": string,
    "stay_seconds": float64,
    "time": string
}
```

Code 3: Formation of browser collections

From the perspective of implementation, browser manufacturers collect data after developer calls *start()*. In Code 3, each time when a user performs an action, including open a new page, switch to another tab or backtrack to former page, will result in a JSON object that contains *device_id* a unique identifier that represents the device, *previous_url* the previous URL of the action, *current_url* the current URL of the action, *stay_seconds* the stay duration of *previous_url* and *time* string of the time of data creation.

7 Discussion

I think; therefore I am.

René Descartes

The proposed APM simultaneously models a sequence of user actions over web browsing and their decision time for each action. It is also designed and conducted from a user study that collects action paths from participants with different browsing behavior. In this chapter, main findings, decisions, and the limitations of this work are discussed.

7.1 Main Findings

Clickstream Modeling The action path model combines an entire action-level clickstream, and the stay duration of each action into an action path encoder. The quantitative results indicate that a simple model can easily classify the existing three types of browsing behaviors (i.e., goal-oriented, fuzzy, and exploring) with 100% of accuracy. The model is also able to universally (cross-user) predict three to five future visit page with given 95% of the browsing context.

Browsing Behaviors and Patterns Three browsing behaviors based on information behavior theory are concluded, which describes three processes of web browsing. The qualitative analysis indicates that the total number of actions are more important for contributing to indication of goal-oriented behavior. The total stay duration and completion efficiency are more important for indicating exploring behavior.

Afterwards, the observed five patterns from the client-side clickstream. The ring and star patterns appear in the fuzzy and exploring tasks. The ring pattern appeared more often in the exploring task, and the star pattern appeared more often in the fuzzy task because differentiating of the information use. A cluster pattern is an indication of individual intent while browsing, and may connects with few hesitation patterns. The overlap pattern discovered in the collected action path has a low overlap ratio, which suggests that action paths tend to be user-specific behavior but may reserve a small region as a common interest in goal-oriented browsing behavior. Next, an analysis based on Ellis' model and Wilson's theory explored the relationship between these patterns and the proposed browsing behaviors. These patterns partially represent a form of browsing behavior. Finally, since the model encodes the entire client-side clickstream and stay duration, the analysis also explains the qualitative reason why the model achieves such a strong performance.

7.2 Decisions

Why the task difficulty is measured by self-rating scale rather than NASA-TLX? NASA-TLX does not provide more insights than the action path regarding APM. As analyzed in Section 5.1, the major purpose of the measurement of task difficulty is to identify inappropriate tasks design (i.e., abnormal outlier) rather than to measure cognitive load by using NASA-TLX. The significant tests of the subjective self-rating scale of task difficulty supports the argument that these tasks are significantly different from one another.

Whether NASA-TLX for cognitive load or self-rating scale for task difficulty is not able to be used in APM since they are impossible to be collected from unseen users in bootstrap phase. Though it is possible to construct a single subjective score as one of the inputs to the action path model, the model learns browsing behaviors from all collected data. This means that if the model is trained based on a dataset with subjective scores, then the dataset is biased by these scores and eventually reduces the generalization ability in a user-independent context.

Why is leave-one-subject-out cross validation (LosoCV) is not applied in classification task? In the case of this thesis, LosoCV is not necessary. Research in the context of human-computer interaction performed a LosoCV for the purpose of claiming a model is tested then it has not previously viewed any unseen user data, This evaluation is a arguably as a representative of a model's bootstrapping performance. However, this is an inappropriate approach for model performance justification in some cases, Kohavi have reviewed the existing model selection techniques [Kohavi, 1995], and have discussed differences between k-Fold CV and LosoCV. For many years, LosoCV has been researched, statistical research [Xu et al., 2012] has proven that LosoCV is asymptotically equivalent to k-Fold cross validation. This verifies the traditional wisdom that the performance of a model that is evaluated by LosoCV tends to worse than k-Fold cross validation because LosoCV increases the variance of generalization error [Bengio and Grandvalet, 2004]. Therefore, Gao et al. have used the model averaging technique (ensemble of multiple models that trained through LosoCV when leaving different subjects) to develop a novel regularization technique [Gao et al., 2016] that helps a model generalize better. Intuitively, when a model that intend to work in a user independent case, The author of the thesis are only interested in how well a model could fit universally, and how the performance could be changed when a model with fixed architecture is applied to more subjects. More precisely, one can observe that LosoCV is essentially trained on a part of a dataset, that is biased to the training process of the model. Therefore, when someone use the best model that gains minimum generalization bound is nothing else but a biased learning with a part of the subjects.

This is not claim that LosoCV is unnecessary in any cases. LosoCV must be applied with consideration of dataset distribution and algorithmic stability. The theoretical insights indicate that LosoCV is critical when a model must be applied in a security context since LosoCV indicates how well a model could interpret highly correlated clusters to individual users and how effective of a model could defend from an unseen attacker.

Bootstrapping is completely trivial and not relevant to the industry because the bootstrap in the context of recommendation (the thesis' application) is valuable if and only if users do not leave the platform after their first arrival. Therefore, one can solve the bootstrapping by giving mainstream selection and mainstream preferences, then provide personalized recommendations after collecting a minimal required dataset because collecting data becomes fairly easy when a user continuously uses a platform.

Why is the experiment designed under three aggregated browsing behaviors instead of using the existing four or more information seeking behaviors? The main reason of aggregating existing information seeking behaviors is to find the best classification ability and expand the task's design scope. The author argue that in an intelligent system, using machines to acts as human beings must be precise enough. Otherwise, doing so will reduce the user's motivation for using an intelligent system since it mis-acts fallible human behavior. Therefore, the author expect the system to work extremely accurately in any cases for a simple classification task. From the perspective of a task design scope, the information seeking behaviors on the web are concluded in a general scenario for all kinds of websites. Designing a suitable task to characterize browsing behavior in a specific website requires sophisticated thinking and clear formalization of all stages that separates the two different behaviors. The boundary of existing behaviors are not qualitatively defined and browsing behavior can be assigned in multiple categories simultaneously.

For instance, in Choo's theory [Choo et al., 1999], web browsing behaviors are categorized into four aspects: formal search, conditioned viewing, informal search and undirected viewing. The formal search and undirected viewing are similar to goal-oriented and exploring behaviors, which discriminate between two extremes of web browsing. However, informal search and conditioned viewing was described by "a good-enough search is satisfactory" and "browse in pre-selected sources" respectively. The fuzziness of "good-enough search", "satisfactory" and "browse in pre-selected" is not clear enough and is subjectively concluded. This fuzziness in different categories of browsing behavior is magnified in Johnson's patterns [Johnson, Ross, 2017]. Therefore, the author mix the browsing behaviors of goal-oriented and exploring behaviors as an individual fuzzy behavior to avoid this uncertainty in the task's design.

7.3 Limitations and Future Works

Lack of data This thesis has a limitation of the lack of data. Though the thesis collected 189 clickstreams from 21 subjects, however, comparing to the baseline action path model with 90323 parameters in Chapter 5, the dataset is still a small dataset for the training and learning task. Moreover, the validation loss (in Figure 5.3) suggests APM remains large capacity to learn more categories of browsing behavior and prediction performance may be improved via reparametrization (in Figure 5.5), it is still fascinating to see the performance of APM on a large dataset, moreover, how this model can adapts to more information on the web, such as the topic of a page, and interpret more detail with attention mechanism.

Data collection This work simulates three proposed browsing behavior through carefully designed browsing tasks. This method only limit to a small group of users, which is not an appropriate approach for a large dataset collection. The author planned to conduct a field study that installs a clickstream collector during a week, however, there are only two subjects after the lab study are willing to participate in the field study.

Reinforcement learning approach As described in Chapter 3, the dataset that applies to APM is an action-level dataset, which means the sequence of URLs is necessarily a series of user actions. This could inspire us to use reinforcement learning approach to train an agent that could explore and learn the environment of the web. Eventually, the agent will be able to learn and optimize the experience of browsing on the web, which implicitly solves the problem of data collection and the lack of supervised data.

Privacy This work monitors an action level of clickstream, which stores all browsing history of a person on a third-party database, and hence brings a trust and privacy issue of the application. The author positively argue that this is a trust issue between users and service providers. As discussed in Chapter 6, browser providers collect the data anonymously, and users use the browser because of trusts, then world wide web consortium formalizes a standardized web API to developers for using this information, and as a browser user can either authorize developers to use this API or give an explicit rejection.

Proactive serving We are in the era that intelligent system surrounding us. The way we interact with an intelligent system is not as natural as we interact with other people. Communications or interactions between humans in a context does not require any trigger word, and a person can brush out a needs or reacts to another immediately. The action path prediction gives a working example that shows proactive serving is possible if we monitor the environment of web browsing. Therefore, it is interesting to study how a user could use this feature and how users react to the elimination of interaction trigger of an intelligent system.

8 Conclusions

Every age has its own myths and
calls them higher truths.

Anonymous

This thesis proposes the action path model that describes client-side user clickstreams. To justify the model, the thesis designed nine browsing tasks for three qualitatively discussed types of browsing behavior based on the theory of information behavior. Then, a user study is held for these tasks that simulated those behaviors. Afterwards, the thesis applied the data collected from user study to the action path model and analyzed the model performance for this data with comparisons to the traditional machine learning approach. The thesis also provided the visualization of these data and discovered the common, individual, and intersection patterns among the client-side clickstream. As an application showcase, a browser plugin was illustrated, which monitors client-side user clickstreams to predict future movements of web browsing. The benefits and drawbacks of this plugin was also discussed in the corresponding chapter. Furthermore, a generic architecture communication flow, and the possibilities of standardization as browser Web APIs are presented for other developers.

The follows summarizes the answers to the research questions:

Understanding (a) A client-side collected clickstream is different from the server-side collected clickstream because of parallel visiting and multiple website visiting. The three types of suggested browsing behaviors are goal-oriented, fuzzy and exploring behaviors. (b) The number of actions, total stay duration and completion efficiency cannot provide an accurate classifier for these three behaviors. However, the number of actions is more important than the others for indicating goal-oriented browsing behavior and the other two features are more important for indicating exploring behavior; (c) The observed patterns in the action path, which include cluster, hesitation, ring, star and overlap contribute to different browsing behaviors; (d) Action paths visually tend to be user-specific but remain common interests in goal-oriented behaviors.

Classification the proposed action path model is 100% accurate for the classification of the three types of browsing behavior, which is trained on a user-independent dataset.

Prediction prediction of three to five future steps can be accurately (>60%) predicted in a simplest action path model.

The findings are generic. The model is an action-level model that models a sequence of user actions and the time of decision making (stay duration). This means that it can be used on desktop, and also can be implemented in context of mobile devices, or even outside the context of web browsing. Similar to other user behavior data, a client-side user clickstream or user actions directly indicate movements of a user and how they make decisions. Understanding, interpreting and predicting these data not only improves the user experience of web browsing, but also useful to help users to reduce useless browsing, which manages their time more efficiently. Moreover, standardizing the data processing process can formalize this feature for developers, which will help them to use the behavior predictions to improve the user experience of their products.

Traditional server collected clickstream data has proved its high value in many fields. This work exposit the value one-step forward, and contributes to models and approaches that hope to bring ponderable research to the community and industry.

Appendix

All resources relates to the thesis are open source, they can be found publicly in [2]:

- Thesis homepage: `https://changkun.us/thesis/`;
- GitHub repostory: `https://github.com/changkun/MasterThesisHCI/`.

All related text, picture and video content are licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License[3]. The other parts of the thesis (such as program source code) are licensed under a MIT Public License [4].

A Content of enclosed CD

1. */docs/* - Documents regarding scheduling and discussion during the thesis
2. */experiments/* - Raw user study designs, raw datasets collected from field study, pilot study and lab study in the thesis. Besides, analysis code to the collected dataset are located in this folder.
3. */keynotes/* - The raw keynote files of thesis commencement and defence presentation slides.
4. */src/* - Developed applications. This folder contains four applications that produced in the thesis: *crawler* is a web spider that collects then entire link relationships in medien.ifi.lmu.de; *gink* is a website that reponsible for crowdsourcing labeling tasks in the wild; *mortal* is the developed web plugin that mentioned in the chapter of application, it has a microservice server and three browser plugin derivatives including lab study collector, field study collector and browsing predictor;
5. */thesis/* - The LaTeXsource code of the thesis, as well as a compiled PDF version.
6. */LICENSE* - An MIT License to all enclosed source code in the CD
7. */README.md* - A brief description of the content enclosed in the CD

[2]The contents found from these links may be revised for improvements that slightly differ from contents from enclosed CD.

[3]`http://creativecommons.org/licenses/by-nc-sa/4.0/`

[4]`https://github.com/changkun/MasterThesisHCI/blob/master/LICENSE`

B Tasks and Questionnaire in Lab Study

B.1 Phase 1: Browsing Task

This section approximately takes 80 minutes.

In this study, you are asked to accomplish a series of tasks provided in the table below. Please read the following tips carefully before you do the task [5].

1. **Please start from the given starting page.** You can then visit any other page. For instance, if you find a task too difficult, you can visit any other websites that help you accomplish the task (e.g. Google as a search engine), but you should only use the browser.

2. The tasks are designed to take **5 10 minutes**. Do not feel stressed if you spend more time because you have 80 minutes in total to **do the 9 tasks**. You will be notified if you spend more than 10 minutes on a task. You can decide to go to the next task or spend some to accomplish the unfinished task.

3. **Close the browser before you start working on the next task.**

4. **Unfortunately, questions cannot be answered while doing the tasks. Please ask them before starting a task if something is not clear.**

B.1.1 Task Group 1: Amazon.com

Task Category: Shopping

1. Assume your smartphone was broken and you have 1200 euros as your budget. You want to buy an iPhone, a protection case, and a wireless charging dock. Look for these items and add them to your cart.

   Requirement to Finish: Click "Proceed to checkout" when you finished, exit the browser when you see the "sign in" page.

2. You want to buy a gift for your best friend as a birthday present. Add three items to your cart as candidate.

   Requirement to Finish: Click "Proceed to checkout" when you finished, exit the browser when you see the "sign in" page.

3. Look for a product category that you are interested in and start browsing. Add three items to your cart that you would like to buy.

   Requirement to Finish: Clicked "Proceed to checkout" when time is up, exit the browser when you see the "sign in" page.

How difficult was the task? (1 5, 1 means very easy, 5 means very difficult)

———, ———, ———

B.1.2 Task Group 2: Medium.com

Task Category: Media

[5] The order of the tasks are rearranged through Latin square, this section only illustrate one possible order of tasks

1. Assume you were making plans for your summer vacation. You want to visit Tokyo, Kyoto, and Osaka. You want to find out what kind of experience other people made when traveling to these three places in Japan. Your task is to find three posts for traveling tips regarding these cities. Elevate a post if it is one of your choices.

   Requirement to Finish: Write down three tips. Close the browser when you are finished.

2. Assume you got an occasion to visit China for business. You are free to travel to China for a week. You want to make a travel plan for touring China within a week. Your task is to find out what kind of experience other how people made when going to secondary cities or towns in China, then decide on three cities you want to visit (excluding Beijing, Shanghai, Guangzhou, and Shenzhen). Elevate if a post helped you make a decision.

   Requirement to Finish: Write down the names of the cities you decided. Close the browser when you are finished.

3. Visit a category you are interested in and elevate the post you like.

   Requirement to Finish: Close the browser when time is up.

How difficult was the task? (1 5, 1 means very easy, 5 means very difficult)
______, ______, ______

B.1.3 Task Group 3: Dribbble.com

Task Category: Design

1. You are hired to a Cloud Computing startup company. You get an assignment to designing the logo of the company. Search for existing logos for inspiration and download three candidate logos you like the most.

   Requirement to Finish: Close the browser when you finished the download.

2. You are preparing a presentation and need one picture for each of these animals: cat, dog, and ant. Download the three pictures you like the most.

   Requirement to Finish: Close the browser when you finished the download.

3. Explore dribbble and download images you like the most while you browse.

   Requirement to Finish: Close the browser when you finished the download.

How difficult was the task? (1 5, 1 means very easy, 5 means very difficult)
______, ______, ______

B.2 Phase 2: Questionnaire

This section approximately takes 10 minutes.

1. Age: ______

2. Gender: Female / Male

3. What is your study program or occupation?

4. What are the websites that you access mostly? List your top-5 (max 10, including private use).

5. What do you usually do when you access these websites? Shortly answer your case for all the websites you listed in above and name two common reasons, ordered by frequency. (For example, for YouTube, the most common reason could be "Just for fun", the second most common reason "Looking for tutorial". Then write as "Mostly for fun, sometimes for learning" below.)
6. Do you use bookmarks to save webpages that you have found through a search engine? If so, why?
7. Which browser do you use mainly on your PC or Mac? Chrome / Safari / IE / Microsoft Edge / Firefox / Others, the name is: ______
8. Would you like to participate in a follow-up study? The study will ask you to install a browser plugin for a week which anonymously records your browsing history. Yes / No
9. Do you have any feedback on this questionnaire?

B.3 Unselected Tasks

This section lists all designed tasks but unselected to lab study.

B.3.1 Goal-oriented Task

1. **www.github.com**: You are comparing three most popular frontend desktop frameworks: Electron / NW.js / ReactNative Desktop. Your goal is to find out the latest release download link.
2. **www.medien.ifi.lmu.de**: You are a fresh medieninformatik student major in HCI program. You wants to find out recommended first semester study plan provided by the program, then select "Human-Computer Interaction II" opened in WS18/19 and check previous "Human-Computer Interaction I" opened in SS18 and SS17.
3. **www.en.uni-muenchen.de**: You are a international student who want to apply economics program for your master study at LMU. Find the page for application requirement.
4. **www.ielts.org**: You live in Munich, you want to participate to IELTS test next year on Feburary. Looking for the entrace to register the examination. You must keep seeking and stop when you selected the first track of Feburary test.
5. **www.bloomberg.com**: You somehow heared about Bloomberg reported a news about China use tiny chips infiltrate U.S companies. You wants to find the article.
6. **www.reddit.com**: You are a fan of Marvel comics, you want to view some spoilers regarding a comming moive "The Avengers 4". Find latest three post that spoilers The Avengers 4.
7. **www.facebook.com**: You are a facebook user, and you have a wide social. However you don't wants to see parenting information in your timeline, you wish to turn them off for a year from your timeline; then recently you start interested in ping pong, you want to join a related local group.
8. **www.twitter.com**: You lost your phone and phone number, and you bought a new one. However the old phone number was registered in your twitter account, you want to change it for your account safety. Please find the entrace to change your phone number and password. Then you becomes curious on twitter's settings. You want to know how twitter use your data and prevent twitter collect your data.

9. **www.youtube.com**: You want to be a Youtuber. You wants to know how to earn money from making videos, and what should you concern when you publishing a video.

10. **www.google.com**: You can't access your gmail. You want to findout whether gmail are current malfuntioning or not. Contact instance messaging support.

B.3.2 Fuzzy Task

1. **www.github.com**: You were a senior developer. Your boss wants you write a report regarding the tends of current development techniques. You want to find the most three popular (top-3 stars) web backend Go frameworks and access their repository, write their name down on a paper when you decided.

2. **www.medien.ifi.lmu.de**: You are a fresh medieninformatik student. You wants to select three lectures, one seminar and one practicum for your study in WS18/19.

3. **www.arxiv.org**: Find the most recent published a overview paper for these three topics respectively: affective computing, convolutional neural networks, distributed consistency algorithm.

4. **www.google.com**: You want to know how google profiling you based on your history. Find your personality profile that created by Google.

5. **www.bloomberg.com**: You want to find the relevant news regarding the progress of China use tiny chips infiltrate U.S companies.

B.3.3 Exploring Task

1. **www.github.com**: Browsing github and select three github repository your most interested in.

2. **www.medien.ifi.lmu.de**: Browsing the website until time is up.

3. **www.en.uni-muenchen.de**: Browsing the website until time is up.

4. **www.ielts.org**: Browsing the website to see what you can do except register to examination.

5. **www.bloomberg.com**: Browsing the website until time is up.

6. **www.reddit.com**: Browsing the website until time is up.

7. **www.facebook.com**: Browsing the website until time is up.

8. **www.twitter.com**: Browsing the website until time is up.

9. **www.youtube.com**: Browsing the website until time is up.

10. **www.arxiv.org**: Browsing the website for categories you interested in until time is up.

11. **www.google.com**: Browsing google until time is up.

C Raw Data Illustration

C.1 Subjective Difficulty Score from Lab Study

Table C.1 illustrates the raw subjective difficulty score from all of the participants.

Table C.1: Subjective task difficulty from lab study

Subject ID	Amazon.com	Medium.com	Dribbble.com
0	2, 1, 2	2, 4, 1	2, 3, 2
1	2, 2, 1	2, 3, 1	1, 5, 1
2	3, 2, 2	2, 5, 3	3, 1, 3
3	3, 4, 2	2, 5, 2	3, 3, 2
4	2, 1, 3	3, 5, 3	2, 1, 3
5	2, 2, 1	3, 4, 1	1, 3, 2
6	3, 4, 2	3, 5, 3	4, 3, 2
7	1, 1, 1	3, 5, 2	2, 1, 1
8	2, 3, 2	2, 5, 2	3, 1, 1
9	1, 3, 2	2, 3, 2	2, 3, 3
10	2, 2, 3	1, 4, 5	1, 2, 3
11	3, 2, 1	3, 4, 1	3, 2, 2
12	4, 1, 3	5, 4, 2	2, 2, 1
13	2, 2, 2	2, 3, 1	2, 2, 1
14	5, 1, 3	2, 4, 1	4, 2, 3
15	1, 2, 1	1, 3, 1	1, 1, 1
16	3, 1, 1	3, 4, 3	2, 2, 3
17	2, 2, 1	2, 3, 1	3, 2, 2
18	3, 2, 2	2, 2, 1	1, 1, 2
19	1, 3, 2	3, 5, 1	2, 3, 2
20	3, 3, 2	3, 5, 4	2, 3, 5

C.2 Raw clickstream data

Code 4 is an illustration of the collected clickstream data. It intends to help readers to have better understanding of this thesis. The complete dataset can be found in the enclosed CD.


```

[
    {
        "task_id": 1,
        "clickstream": [
            {"user_id":1,"previous_url":"","current_url":"https://www.amazon.com/","stay_seconds":26.214,"time":"2018-12-03T19:44:19Z"},
            {"user_id":1,"previous_url":"https://www.amazon.com/","current_url":"https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps\u0026field-keywords=iphone","stay_seconds":10.712,"time":"2018-12-03T19:54:19Z"},
            {"user_id":1,"previous_url":"https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps\u0026field-keywords=iphone","current_url":"https://www.amazon.com/s/ref=nb_sb_noss?url=node%3D7072561011\u0026field-keywords=iphone+xs\u0026rh=n%3A7072561011%2Ck%3Aiphone+xs","stay_seconds":6.099,"time":"2018-12-03T19:54:25Z"},
            ...
```

```
            {"user_id":1,"previous_url":"https://www.amazon.com/gp/product/handle-buy-box/ref=dp_start-bbf_1_glance","current_url":"https://www.amazon.com/gp/huc/view.html?ie=UTF8\u0026increasedItems=C788d76cc-7a30-44cc-8041-85993f4d6716\u0026newItems=C788d76cc-7a30-44cc-8041-85993f4d6716%2C1","stay_seconds":10.282,"time":"2018-12-03T19:57:40Z"},
            {"user_id":1,"previous_url":"https://www.amazon.com/gp/huc/view.html?ie=UTF8\u0026increasedItems=C788d76cc-7a30-44cc-8041-85993f4d6716\u0026newItems=C788d76cc-7a30-44cc-8041-85993f4d6716%2C1","current_url":"https://www.amazon.com/gp/cart/view.html/ref=lh_cart_vc_btn","stay_seconds":1.886,"time":"2018-12-03T19:57:41Z"},
            {"user_id":1,"previous_url":"https://www.amazon.com/gp/cart/view.html/ref=lh_cart_vc_btn","current_url":"https://www.amazon.com/ap/signin","stay_seconds":71.552,"time":"2018-12-03T19:58:53Z"},
        ]
    },
    {
        ...
    },
    ...
]
```

Code 4: Formation of browser collections

Bibliography

References

[Aceto et al., 1994] Aceto, S., Delrio, C., Dondi, C., Fischer, T., Kastis, N., Klein, R., Strauss, A., and Corbin, J. (1994). Grounded theory methodology: An overview. *Handbook of qualitative research. Thousand Oaks: Sage Publications*.

[Amo Filvá et al., 2018] Amo Filvá, D., Alier Forment, M., García Peñalvo, F. J., Fonseca Escudero, D., and Casany Guerrero, M. J. (2018). Learning analytics to assess students behavior with scratch through clickstream. In *Proceedings of the Learning Analytics Summer Institute Spain 2018: León, Spain, June 18-19, 2018*, pages 74–82. CEUR-WS. org.

[Baumann et al., 2018] Baumann, A., Haupt, J., Gebert, F., and Lessmann, S. (2018). The price of privacy: An evaluation of the economic value of collecting clickstream data. *Business and Information Systems Engineering*.

[Benevenuto et al., 2009] Benevenuto, F., Rodrigues, T., Cha, M., and Almeida, V. (2009). Characterizing user behavior in online social networks. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, IMC '09, pages 49–62, New York, NY, USA. ACM.

[Bengio and Grandvalet, 2004] Bengio, Y. and Grandvalet, Y. (2004). No unbiased estimator of the variance of k-fold cross-validation. *Journal of machine learning research*, 5(Sep):1089–1105.

[Brodwin, D., D. O'Connell, and M. Valdmanis., 1995] Brodwin, D., D. O'Connell, and M. Valdmanis. (1995). Mining the Clickstream. pages 101–106.

[Bucklin and Sismeiro, 2000] Bucklin, R. E. and Sismeiro, C. (2000). How sticky is your web site? modeling site navigation choices using clickstream data. Technical report, Working paper, Anderson School UCLA.

[Carr, 2000] Carr, N. G. (2000). Hypermediation: commerce as clickstream. *Harvard Business Review*, 78(1):46–47.

[Cavoukian, 2000] Cavoukian, A. (2000). Should the oecd guidelines apply to personal data online. In *A report to the 22nd international conference of data protection commissioners*.

[Chatterjee, Patrali and Hoffman, Donna L and Novak, Thomas P, 2003] Chatterjee, Patrali and Hoffman, Donna L and Novak, Thomas P (2003). Modeling the clickstream: Implications for web-based advertising efforts. *Marketing Science*, 22(4):520–541.

[Chi et al., 2017] Chi, Y., Jiang, T., He, D., and Meng, R. (2017). Towards an integrated clickstream data analysis framework for understanding web users' information behavior. *iConference 2017 Proceedings*.

[Cho et al., 2014] Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

[Choo et al., 1999] Choo, C. W., Detlor, B., and Turnbull, D. (1999). Information seeking on the web: An integrated model of browsing and searching.

[Cochran and Cox, 1950] Cochran, W. G. and Cox, G. M. (1950). Experimental designs.

[Courtheoux, 2000] Courtheoux, R. J. (2000). Database marketing connects to the internet. *Interactive Marketing*, 2(2):129–137.

[Dijkstra, 1959] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.

[Ellis, 1989] Ellis, D. (1989). A behavioural model for information retrieval system design. *Journal of information science*, 15(4-5):237–247.

[Ellis et al., 1993] Ellis, D., Cox, D., and Hall, K. (1993). A comparison of the information seeking patterns of researchers in the physical and social sciences. *Journal of documentation*, 49(4):356–369.

[Ellis and Haugan, 1997] Ellis, D. and Haugan, M. (1997). Modelling the information seeking patterns of engineers and research scientists in an industrial environment. *Journal of documentation*, 53(4):384–403.

[Fisher and Julien, 2009] Fisher, K. E. and Julien, H. (2009). Information behavior. *Annual Review of Information Science and Technology*, 43(1):1–73.

[Friedman, Wayne and Weaver, Jane, 1995] Friedman, Wayne and Weaver, Jane (1995). Calculating cyberspace: tracking “clickstreams.”.

[Gao et al., 2016] Gao, Y., Zhang, X., Wang, S., and Zou, G. (2016). Model averaging based on leave-subject-out cross-validation. *Journal of Econometrics*, 192(1):139 – 151.

[Garg and Agarwal, 2019] Garg, A. and Agarwal, M. (2019). Machine translation: A literature review. *CoRR*, abs/1901.01122.

[Giannini, 1998] Giannini, T. (1998). Information receiving: A primary mode of the information process. *Proceedings of the ASIS Annual Meeting*, 35.

[Gindin, 1997] Gindin, S. E. (1997). Lost and found in cyberspace: Informational privacy in the age of the internet. *San Diego L. Rev.*, 34:1153.

[Goldfarb, 2002] Goldfarb, A. (2002). Analyzing website choice using clickstream data. In *The Economics of the Internet and E-commerce*, pages 209–230. Emerald Group Publishing Limited.

[Graves, 2012] Graves, A. (2012). Sequence transduction with recurrent neural networks. *CoRR*, abs/1211.3711.

[Gundala and Spezzano, 2018] Gundala, L. A. and Spezzano, F. (2018). Readers’ demanded hyperlink prediction in wikipedia. In *Companion Proceedings of the The Web Conference 2018*, WWW ’18, pages 1805–1807, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–80.

[Huang et al., 2012] Huang, J., Lin, T., and White, R. W. (2012). No search result left behind: branching behavior with browser tabs. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 203–212. ACM.

[Huang and White, 2010] Huang, J. and White, R. W. (2010). Parallel browsing behavior on the web. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, pages 13–18. ACM.

[Johnson, Ross, 2017] Johnson, Ross (2017). Website Browsing Behavior Patterns. `https://3.7designs.co/blog/2017/10/website-browsing-behavior-patterns`. Accessed: 2018-12-29.

[Jozefowicz et al., 2015] Jozefowicz, R., Zaremba, W., and Sutskever, I. (2015). An empirical exploration of recurrent network architectures. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 2342–2350. JMLR.org.

[Kammenhuber et al., 2006] Kammenhuber, N., Luxenburger, J., Feldmann, A., and Weikum, G. (2006). Web search clickstreams. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, IMC '06, pages 245–250, New York, NY, USA. ACM.

[Kang, 1997] Kang, J. (1997). Information privacy in cyberspace transactions. *Stan. L. Rev.*, 50:1193.

[Kohavi, 1995] Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada.

[Lin et al., 2012] Lin, M., Lin, M., and Kauffman, R. J. (2012). From clickstreams to searchstreams: Search network graph evidence from a b2b e-market. In *Proceedings of the 14th Annual International Conference on Electronic Commerce*, ICEC '12, pages 274–275, New York, NY, USA. ACM.

[Liu et al., 2010] Liu, C., White, R. W., and Dumais, S. (2010). Understanding web browsing behaviors through weibull analysis of dwell time. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 379–386. ACM.

[Lori Lewis, 2017] Lori Lewis (2017). What Your Audience Is Doing When They're Not Listening To You. `https://www.allaccess.com/merge/archive/26034/what-your-audience-is-doing-when-they-re-not`. Accessed: 2018-12-28.

[Lori Lewis, 2018] Lori Lewis (2018). What Happens In An Internet Minute: 2018 Update. `https://www.allaccess.com/merge/archive/28030/2018-update-what-happens-in-an-internet-minute`. Accessed: 2018-12-28.

[Lourenço and Belo, 2006] Lourenço, A. G. and Belo, O. O. (2006). Catching web crawlers in the act. In *Proceedings of the 6th International Conference on Web Engineering*, ICWE '06, pages 265–272, New York, NY, USA. ACM.

[Lyons and Henderson, 2005] Lyons, B. and Henderson, K. (2005). Opinion leadership in a computer-mediated environment. *Journal of Consumer Behaviour: An International Research Review*, 4(5):319–329.

[Mandese, 1995] Mandese, J. (1995). Clickstreams' in cyberspace. *Advertising Age*, 66(12):18–18.

[Mann and Whitney, 1947] Mann, H. B. and Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60.

[Meier and Elsweiler, 2016] Meier, F. and Elsweiler, D. (2016). Going back in time: An investigation of social media re-finding. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 355–364, New York, NY, USA. ACM.

[Mikolov et al., 2013a] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

[Mikolov et al., 2013b] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

[Mobasher et al., 2001] Mobasher, B., Dai, H., Luo, T., and Nakagawa, M. (2001). Effective personalization based on association rule discovery from web usage data. In *Proceedings of the 3rd International Workshop on Web Information and Data Management*, WIDM '01, pages 9–15, New York, NY, USA. ACM.

[N and Ravindran, 2018] N, C. T. and Ravindran, B. (2018). A neural attention based approach for clickstream mining. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, CoDS-COMAD '18, pages 118–127, New York, NY, USA. ACM.

[Novick, Bob, 1995] Novick, Bob (1995). Internet Marketing: The Clickstream. `http:/www.im.com/archives/9503/0375.html`http:/www.im.com/archives/9503/0375.html. Accessed: 2018-12-10.

[Padmanabhan et al., 2001] Padmanabhan, B., Zheng, Z., and Kimbrough, S. O. (2001). Personalization from incomplete data: What you don't know can hurt. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 154–163, New York, NY, USA. ACM.

[Park et al., 2017] Park, J., Denaro, K., Rodriguez, F., Smyth, P., and Warschauer, M. (2017). Detecting changes in student behavior from clickstream data. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, LAK '17, pages 21–30, New York, NY, USA. ACM.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Reagle and Cranor, 1999] Reagle, J. and Cranor, L. F. (1999). The platform for privacy preferences. *Communications of the ACM*, 42(2):48–55.

[Reidenberg, 1996] Reidenberg, J. R. (1996). Governing networks and rule-making in cyberspace. *Emory LJ*, 45:911.

[Reidenberg, 1999] Reidenberg, J. R. (1999). Resolving conflicting international data privacy rules in cyberspace. *Stan. L. Rev.*, 52:1315.

[Reinfelder et al., 2014] Reinfelder, L., Benenson, Z., and Gassmann, F. (2014). *Differences between Android and iPhone Users in Their Security and Privacy Awareness*, pages 156–167.

[Rumelhart et al., 1988] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, Cambridge, MA, USA.

[Sadagopan and Li, 2008] Sadagopan, N. and Li, J. (2008). Characterizing typical and atypical user sessions in clickstreams. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 885–894, New York, NY, USA. ACM.

[Sandoiu, Ana, 2018] Sandoiu, Ana (2018). Do Android and iPhone users have different personalities? https://www.medicalnewstoday.com/articles/314376.php. Accessed: 2019-01-04.

[Schneider et al., 2009] Schneider, F., Feldmann, A., Krishnamurthy, B., and Willinger, W. (2009). Understanding online social network usage from a network perspective. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, IMC '09, pages 35–48, New York, NY, USA. ACM.

[Schonberg et al., 2000] Schonberg, E., Cofino, T., Hoch, R., Podlaseck, M., and Spraragen, S. L. (2000). Measuring success. *Communications of the ACM*, 43(8):53–57.

[Shimada et al., 2018] Shimada, A., Taniguchi, Y., Okubo, F., Konomi, S., and Ogata, H. (2018). Online change detection for monitoring individual student behavior via clickstream data on e-book system. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, LAK '18, pages 446–450, New York, NY, USA. ACM.

[Shires, Glen and Jaegenstedt, Philip, 2018] Shires, Glen and Jaegenstedt, Philip (2018). Web Speech API. https://w3c.github.io/speech-api/. Accessed: 2019-01-03.

[Skok, 1999] Skok, G. (1999). Establishing a legitimate expectation of privacy in clickstream data. *Mich. Telecomm. & Tech. L. Rev.*, 6:61.

[StatCounter, 2018] StatCounter (2018). Usage share of web browsers. http://gs.statcounter.com/browser-market-share#monthly-201811-201811-bar. Accessed: 2018-12-29.

[Sun and Xin, 2017] Sun, Y. and Xin, C. (2017). Using coursera clickstream data to improve online education for software engineering. In *Proceedings of the ACM Turing 50th Celebration Conference - China*, ACM TUR-C '17, pages 16:1–16:6, New York, NY, USA. ACM.

[Sutskever et al., 2014] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

[Suykens and Vandewalle, 1999] Suykens, J. A. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300.

[The Apache Software Foundation, 1995] The Apache Software Foundation (1995). About Apache: How Apache Came to Be. http://httpd.apache.org/ABOUT_APACHE.html. Accessed: 2018-12-10.

[Ting et al., 2005] Ting, I.-H., Kimble, C., and Kudenko, D. (2005). Ubb mining: Finding unexpected browsing behaviour in clickstream data to improve a web site's design. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '05, pages 179–185, Washington, DC, USA. IEEE Computer Society.

[Vassio et al., 2018] Vassio, L., Drago, I., Mellia, M., Houidi, Z. B., and Lamali, M. L. (2018). You, the web, and your device: Longitudinal characterization of browsing habits. *ACM Trans. Web*, 12(4):24:1–24:30.

[W3C, 2018] W3C (2018). WebAssembly Core Specification. https://webassembly.github.io/spec/core/bikeshed/index.html. Accessed: 2019-01-03.

[Walsh, John and Godfrey, Sue, 2000] Walsh, John and Godfrey, Sue (2000). The Internet: a new era in customer service. *European Management Journal*, 18(1):85–92.

[Wang et al., 2017] Wang, G., Zhang, X., Tang, S., Wilson, C., Zheng, H., and Zhao, B. Y. (2017). Clickstream User Behavior Models. *ACM Trans. Web*, 11(4):21:1–21:37.

[Wang et al., 2016] Wang, G., Zhang, X., Tang, S., Zheng, H., and Zhao, B. Y. (2016). 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, New York, NY, USA. ACM.

[Waterson et al., 2002a] Waterson, S., Landay, J. A., and Matthews, T. (2002a). In the lab and out in the wild: Remote web usability testing for mobile devices. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '02, pages 796–797, New York, NY, USA. ACM.

[Waterson et al., 2002b] Waterson, S. J., Hong, J. I., Sohn, T., Landay, J. A., Heer, J., and Matthews, T. (2002b). What did they do? understanding clickstreams with the webquilt visualization system. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '02, pages 94–102, New York, NY, USA. ACM.

[Weller, 2018] Weller, T. (2018). Compromised account detection based on clickstream data. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pages 819–823, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

[Werbos, 1990] Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

[Williams and Zipser, 1989] Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

[Wilson, 1981] Wilson, T. D. (1981). On user studies and information needs. *Journal of documentation*, 37(1):3–15.

[Wilson, 1997] Wilson, T. D. (1997). Information behaviour: an interdisciplinary perspective. *Information processing & management*, 33(4):551–572.

[Xu et al., 2012] Xu, G., Huang, J. Z., et al. (2012). Asymptotic optimality and efficient computation of the leave-subject-out cross-validation. *The Annals of Statistics*, 40(6):3003–3030.

[Yamakami, 2009] Yamakami, T. (2009). Inter-service revisit analysis of three user groups using intra-day behavior in the mobile clickstream. In *Proceedings of the 2009 International Conference on Hybrid Information Technology*, ICHIT '09, pages 340–344, New York, NY, USA. ACM.

[Yang et al., 2014] Yang, Z., Wilson, C., Wang, X., Gao, T., Zhao, B. Y., and Dai, Y. (2014). Uncovering social network sybils in the wild. *ACM Trans. Knowl. Discov. Data*, 8(1):2:1–2:29.

[Zaloudek, 2018] Zaloudek, J. (2018). User Behavior Clustering and Behavior Modeling Based on Clickstream Data. Master's thesis, Czech Technical University in Prague, Faculty of Electrical Engineering Department of Computer Science.

[Zhang et al., 2016] Zhang, X., Brown, H.-F., and Shankar, A. (2016). Data-driven personas: Constructing archetypal users with clickstreams and user telemetry. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5350–5359, New York, NY, USA. ACM.