# Deep Sequential Models for Task Satisfaction Prediction

Rishabh Mehrotra[†], Ahmed Hassan Awadallah[‡], Milad Shokouhi[‡], Emine Yilmaz[†◇], Imed Zitouni[‡], Ahmed El Kholy[‡], Madian Khabsa[‡*]

[†]University College London, London, United Kingdom
[‡]Microsoft Inc., Redmond, WA, United States
[◇]The Alan Turing Institute, British Library, London, United Kingdom
{r.mehrotra,e.yilmaz}@cs.ucl.ac.uk,{hassanam,milads,izitouni,ahkhol,makhab}@microsoft.com

## ABSTRACT

Detecting and understanding implicit signals of user satisfaction are essential for experimentation aimed at predicting searcher satisfaction. As retrieval systems have advanced, search tasks have steadily emerged as accurate units not only to capture searcher's goals but also in understanding how well a system is able to help the user achieve that goal. However, a major portion of existing work on modeling searcher satisfaction has focused on query level satisfaction. The few existing approaches for task satisfaction prediction have narrowly focused on simple tasks aimed at solving atomic information needs.

In this work we go beyond such atomic tasks and consider the problem of predicting user's satisfaction when engaged in complex search tasks composed of many different queries and subtasks. We begin by considering holistic view of user interactions with the search engine result page (SERP) and extract detailed *interaction sequences* of their activity. We then look at query level abstraction and propose a novel deep sequential architecture which leverages the extracted interaction sequences to predict query level satisfaction. Further, we enrich this model with auxiliary features which have been traditionally used for satisfaction prediction and propose a unified multi-view model which combines the benefit of user interaction sequences with auxiliary features.

Finally, we go beyond query level abstraction and consider query sequences issued by the user in order to complete a complex task, to make task level satisfaction predictions. We propose a number of functional composition techniques which take into account query level satisfaction estimates along with the query sequence to predict task level satisfaction. Through rigorous experiments, we demonstrate that the proposed deep sequential models significantly outperform established baselines at both query and task satisfaction prediction. Our findings have implications on metric development for gauging user satisfaction and on designing systems which help users accomplish complex search tasks.

[*]Part of the work was conducted while the first author was visiting Microsoft Research.


*CIKM'17, November 6–10, 2017, Singapore*
 ISBN 978-1-4503-4918-5/17/11…$15.00
DOI: http://dx.doi.org/10.1145/3132847.3133001

## 1 INTRODUCTION

As search systems have advanced, an increasingly larger proportions of users are relying on search engine to satisfy their information needs. Developing better understanding of how users interact with search engines is becoming important for gauging user satisfaction and improving user's search experience. Since obtaining explicit feedback from users is often prohibitively expensive and challenging to implement in real-world systems, commercial search engines have exploited implicit feedback signals derived from user activity. While users interact with a search engine, they leave behind fine grained traces of interaction signals. These interaction signals contain valuable information, which could be useful for predicting user satisfaction as well as developing metrics for search engine evaluation to assist rapid experimentation.

User initiated search is often motivated by a search goal, or a task. A simple task refers to an atomic information need resulting in one or more queries [19]. Understanding and evaluating a search engine's performance from a task centric view attains paramount importance. Most existing work on gauging user satisfaction have focused on query level satisfaction [10, 11, 15, 22], with some initial efforts aimed at measuring task satisfaction for simple tasks [14]. Often, independent information needs arise from an overall complex search task, where a *complex search task* refers to a multi-aspect or a multi-step information need consisting of a set of related tasks, each of which might recursively be complex [2, 19, 32, 33]. While existing work has primarily focused on measuring user satisfaction on simple search tasks, work on understanding and measuring user satisfaction for complex search tasks remains in its infancy.

In this work, we take a comprehensive look at user satisfaction from different levels of abstractions. We begin by investigating query level satisfaction, and propose a deep sequential model which considers holistic view of user's interaction with the search engine result page (SERP), constructs detailed interaction sequences of their activities and leverages such interaction sequences to predict query level satisfaction. In addition to interaction sequences, we consider various different behavior signals (e.g. click features, dwell times) and treat such signals as auxiliary features providing an alternate view of user interactions. We propose a unified multi-view deep model composed of parallel convolutional and recurrent neural networks capable of utilizing both the views of user interactions for predicting query level satisfaction.

Finally, we go beyond query level abstraction and consider the problem of task satisfaction prediction. We propose a novel functional composition model which takes into account user satisfaction

at the query level and the subtask level when making task satisfaction predictions. We present rigorous evaluation of the proposed approach using crowdsourced judgments as well as large scale pseudo-labeled data and demonstrate that the unified multi-view deep sequential model significantly outperforms a number of established baselines at query satisfaction prediction. We additionally show that the proposed deep sequential models are also better at predicting task level satisfaction. Our findings provide a valuable tool for gauging task satisfaction and developing next generation task-aware search engines.

## 2 RELATED WORK

The current research builds upon and advances research in three directions, which we discuss below:

**User Satisfaction:**
The concept of satisfaction was first introduced in IR researches in 1970s according to Su et al. [40]. A recent definition states that "satisfaction can be understood as the fulfillment of a specified desire or goal" [20]. However, search satisfaction itself is a subjective construct and is difficult to measure. Some existing studies tried to collect satisfaction feedback from users directly. For example, Guo *et al.* 's work [10] on predicting Web search success and Feild et al.'s work [5] on predicting searcher frustration were both based on searchers' self-reported explicit judgments. Differently, other researchers employed external assessors to restore the users' search experience and make annotations according to their own opinions [11]. Recently, simplistic user feedback signals have been used to gauge user satisfaction. For instance, it has previously been shown that clicks followed by long dwell times are correlated with satisfaction [7]. Hassan et al. [15] propose to use query reformulation as a negative indicator of search success and thus satisfaction. Kim et al. [21] consider three measures of dwell time and evaluate their use in detecting search satisfaction. Lagun *et al.*[27] consider scroll and viewport features for predicting satisfaction in mobile search.

**Gestures for Relevance & Satisfaction:**
A number of different interaction behaviors have been taken into consideration in the prediction of search user satisfactions including both coarse-grained features (e.g. clickthrough based features in [11]) and fine-grained ones (e.g. cursor position and scrolling speed in [10]). Mouse movement information like scroll and hover have proven to be valuable signals in inferring user behavior and preferences [8, 16, 37], search intent [9], search examination [30] and predicting result relevance [17]. However, none of these studies tried to extract mouse movement patterns and adopt them to predict search satisfaction. Arapakis et al. [3] extracted mouse gestures to measure within-content engagement. Lagun et al. [26] introduced the concept of frequent cursor subsequences (namely motifs) in the estimation of result relevance. User action sequences have been used to predict user satisfaction [14], graded satisfaction [18] and to study search engine switching behavior [38, 42].

**Search Tasks:**
While a major share of prior work has considered search sessions as the focal unit of analysis for seeking behavioral insights, search tasks are emerging as a competing perspective in this space. There have been attempts to extract in-session tasks [19, 31], and cross-session tasks [25, 41] from query sequences based on classification and clustering methods. Kotov et al [25] and Agichtein *et al.* [1] studied the problem of cross-session task extraction via binary same-task classification, and found that different types of tasks demonstrate different life spans. More recently, Mehrotra *et al.* [32] presented a nonparametric clustering model for subtask extraction but ignored task specific as well as coherence based insights.

Our work is different from existing work not only in measuring query level satisfaction but also in measuring task satisfaction. We propose novel ways of combining different views of user interactions in a unified model for predicting query satisfaction. Further, unlike past work which considers simple tasks composed of query reformulations for measuring task satisfaction, we go beyond such simpler tasks and also consider complex tasks composed of many different queries and subtasks.

## 3 PROBLEM FORMULATION

Our goal in this work is to extract and leverage user interaction data to predict query and task level satisfaction. We begin by defining the key concepts used throughout the paper.
**Sequence:** Given a search impression and a list of possible user actions, a sequence is defined as a time-ordered list of actions performed by the user when interacting with the search result page.
**Search Task:** A search task is an atomic information need resulting in one or more queries [19]
**Complex Task:** A complex search task is a multi-aspect or a multi-step information need consisting of a set of related tasks, each of which might recursively be complex [33].
With this background, we formally define the problem of satisfaction prediction:
**Query Satisfaction (QSAT):** Given user interaction information, predict whether the user was satisfied with the search results.
**Task Satisfaction:** Given a sequence of queries issued by the user to accomplish a complex task along with user interaction information for each query, predict user's satisfaction in completing the overall complex task.

In order to make task satisfaction predictions, we leverage query level satisfaction estimates as well as subtask level satisfaction estimates. While few efficient approaches exist for automatically identifying subtasks [32, 33], we assume access to subtask demarcation information (obtained via crowdsourced labeling) for the scope of this work.

We first describe the technique used to extract meaningful action sequences from user interactions with SERP (Section 4) . We then present in Section 5 our proposed deep sequential model for query level satisfaction prediction. Finally, in Section 6 we present different techniques for functional composition of query satisfaction estimates to make task satisfaction predictions.

## 4 EXTRACTING USER INTERACTION DATA

The richness of the result page rendered in response to a user query allows users to interact with SERPs in myriad ways, including clicking results, scrolling, expanding task panels, hovering over

| Action | Description |
|---|---|
| Click_algoX | Click on the X-th algorithmic result |
| Click_Ans | Click on any answer (non-image) result |
| Click_IMG | Click on any image result |
| MouseRead | horizontal line across a result snippet of length > 50px and duration > 100 ms that goes from left to right which starts and ends inside an algo-result, or advertisement or an answer result |
| Scroll | page scroll recorded on the search engine result page |
| Move | any cursor movement of length > 10px and duration greater than > 50 ms |
| pause | smallPause: no cursor movement on the SERP for time < 5 seconds<br>mediumPause: no cursor movement on the SERP for 5s < time < 20s<br>longPause: no cursor movement on the SERP for 20s < time < 40s<br>veryLongPause: no cursor movement on the SERP for time > 40s |
| Resize | change in the size of the window/screen encompassing the result page |
| IssueQuery | user movement to the Search Box on the SERP and typing of text in the query box |
| dwellTime | smallDwellTime: dwell time on a clicked result URL with time spent < 10s<br>mediumDwellTime: dwell time on a clicked result URL with 10s < time < 40s<br>longDwellTime: dwell time on a clicked result URL with time spent > 40s |
| QuickBack | click on a SERP URL followed by returning back to the SERP within 5s |

**Table 1: Examples of actions considered along with their description used to create the user interaction sequence.**

| Example Sequence |
|---|
| Scroll → smallPause → Move-algo-1 → smallPause → Move-algo-2 → smallPause → Click-algo-2 |

**Table 2: Example of sequences extracted.**

images, pausing to read and absorb content among others. Following Mehrotra *et al.* [34], we extract interaction sequence from user interaction with the SERP. To do so, we construct a universal action sequence timeline from the following three different timelines:

(1) **Viewport Timeline**: Viewport is defined as the position of the webpage that is visible at any given time to the user. Viewport timeline allows us to consider user actions concerning the viewport, for example, scroll on the result page and resize of the screen.
(2) **Cursor Timeline**: The cursor timeline provides us with all the cursor related user activity. Backend search logs record detailed user mouse activity which helps us to track the mouse movement and link the corresponding cursor activity to the different elements on the SERP.
(3) **Keyboard Timeline**: The keyboard timeline records all keyboard related user activity (for example, text enter).

For each search impression, we log the three timelines with corresponding user actions along with the timestamp. Based on these three timelines, we generate one holistic universal action sequence timeline describing all user activity on the SERP by temporal sorting of individual timelines followed by stacking up the three timelines, and then interleaving them based on timestamps of the recorded actions. This provides us with a universal sequence of user interaction, examples of which are shown in Table 2. We next take a more detailed look at the actions considered to construct the timelines.

**Actions Considered**: A number of actions were considered which include all types of interactions performed by the users. Table 1 lists the major actions considered. For details on the actions considered, the interested is referred to Mehrotra *et al.* [34].

## 5 QUERY LEVEL SAT PREDICTION

While implicit feedback measures like mouse clicks, reading and dwell times, gaze tracking have been extensively used in predicting search satisfaction, they ignore the sequence information accompanying any user interaction. Given the detailed action sequence extracted from user's interaction with SERP, we aim at predicting user satisfaction using the extracted sequence.

Gauging user satisfaction is the problem of predicting satisfaction label given a query $q$, the search results page rendered, detailed user interaction actions recorded (Section 4) and aggregate implicit signals according to a parametric probability measure:

$$y = arg\ max_{y \in 0,1} p(y|q, SERP, \mathbf{A}, \mathbf{I}; \theta) \quad (1)$$

where $\theta$ represent a vector of all parameters to learn, q is the query, **A** is the user action sequence and **I** are the implicit signals observed. In order to predict query level satisfaction, we leverage interaction sequences and propose a deep sequential model to predict satisfaction. Further, we augment the sequence model with SERP level signals which have been traditionally used to propose a coupled model which combines interaction sequence information with auxiliary implicit feedback signals to propose a unified model for query level satisfaction prediction.

### 5.1 Sequential Model for SAT

To leverage the entire interaction sequence we make use of recent advancements in the field of deep recurrent network and formulate our problem as that of sequence classification. Recurrent neural networks (RNNs) are a powerful family of connectionist models that capture time dynamics via cycles in the graph, thereby enabling them to process sequences of data. A RNN maintains a memory based on history information, which enables the model to predict the current output conditioned on long distance features. An important characteristic of user interactions is that the resulting sequences are of variable length. Long Short-Term Memory (LSTM) networks are a special case of Recurrent Neural Networks (RNNs) which are capable of creating internal cell states of the network

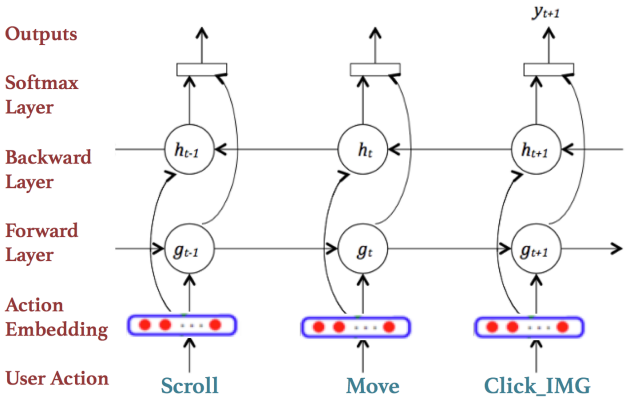


**Figure 1: The Bi-directional LSTM model for query SAT prediction.**

which allow it to exhibit dynamic temporal behavior thereby enabling the RNN to process arbitrary sequences of inputs such as user interaction sequences.

The action-LSTM takes as input a sequence of user actions $x = (x_1, x_2, ...., x_T)$ and computes the hidden sequence $h = (h_1, h_2, ..., h_T)$ as well as the output vector $y = (y_1, y_2, ..., y_T)$ by iterating from $t = 1$ to $T$:

$$h_t = \mathcal{H}(\mathbf{W}_{xh}x_t + \mathbf{W}_{hh}h_{t-1} + b_h) \quad (2)$$

$$y_t = \mathbf{W}_{hy}h_t + b_y \quad (3)$$

where T is the total number of sequences; $\mathbf{W_{xh}}$ are the weight matrices between the input layers $a$ and $h$ and so on; $b$ is a bias vector, and $\mathcal{H}$ is the composite function. The action-LSTM architecture is composed of two components: (i) Action Embeddings and (ii) LSTM sequence model. We next discuss both these components in detail.

**Action Embeddings:**
The input to the action-LSTM is the sequence of user actions on the rendered SERP. While one-hot vector representations have been traditionally used as input to the recurrent neural networks, recently embeddings have shown enhanced performance. We learn action embeddings from the interaction sequence data. Given the set of action sequences, the first layer embeds each action into a continuous vector space using a skip-gram model [35]. Since the input sequences are of arbitrary length, we mask the input sequences with dummy symbol which are ignored during training phase. The embedding layer is optimized jointly with the rest of the model through backpropagation, [12] which results in the model optimizing the individual actions' embedding vectors to be more reflective of their closeness to other actions.

**Sequence LSTM Model:**
After passing through the embedding layer, the input action sequences are input to the LSTM module. The LSTM composite function forming the LSTM cell with peephole connections is defined as:

$$\mathbf{i_t} = \sigma(\mathbf{W}_{xi}x_t + \mathbf{W}_{hi}h_{t-1}) \quad (4)$$

$$\mathbf{f_t} = \sigma(\mathbf{W}_{xf}x_t + \mathbf{W}_{hf}h_{t-1}) \quad (5)$$

$$\mathbf{c_t} = \mathbf{f_t} \odot \mathbf{c}_{t-1} + i_t \odot tanh(\mathbf{W}_{xc}x_t + \mathbf{W}_{hc}h_{t-1}) \quad (6)$$

$$\mathbf{o_t} = \sigma(\mathbf{W}_{hx}x_t + \mathbf{W}_{ho}h_{t-1} + \mathbf{W}_{co}\mathbf{c}_t) \quad (7)$$

$$\mathbf{h_t} = \mathbf{o_t} \odot tanh(\mathbf{c}_t) \quad (8)$$

where $\sigma$ denotes the sigmoid function, $\sigma(z) = (1 + e^{-z})^{-1}$. The superscripts $^{(t)}$ denote the index of the current time step, $\mathbf{i}$, $\mathbf{f}$ and $\mathbf{o}$, are respectively the input, forget and output gates, and $\mathbf{c}$ the cell activation vector with the same size than the hidden vector $\mathbf{h}$. The weight matrices $\mathbf{W}$ from cell $\mathbf{c}$ to gates $\mathbf{i}$, $\mathbf{f}$ and $\mathbf{o}$, are diagonal, and thus, an element $e$ in each gate vector receives only the element $e$ from the cell vector.

In any action in an interaction sequence, we not only have historic actions, but also have future actions user took on the SERP. For many sequence labelling tasks it is beneficial to have access to both past (left) and future (right) actions contexts. However, the LSTM's hidden state $h_t$ takes information only from past, knowing nothing about the future. To leverage future action information, we use bi-directional LSTM (BLSTM) wherein the basic idea is to present each sequence forwards and backwards to two separate hidden states to capture past and future information, respectively. It is important to note that our goal is retrospective satisfaction prediction, i.e., offline prediction of user satisfaction based on the observed interaction signals. While future action sequences will not be available in an online setting, this restriction does not apply in our offline setting, as a result, bi-directional LSTMs can be used in retrospective offline satisfaction prediction. This type of RNN feeds to a same output layer fed forwarded inputs through the two hidden layers. Therefore, the BLSTM computes both forward hidden sequence $\vec{h}$ and backward sequence $\overleftarrow{h}$ as well as the output vector y, by iterating $\vec{h}$ from t = 1 to T , and $\overleftarrow{h}$ from t = T to 1:

$$\vec{h_t} = \mathcal{H}(\mathbf{W}_{x\vec{h}}x_t + \mathbf{W}_{\vec{h}\vec{h}}\vec{h}_{t-1} + b_{\vec{h}}) \quad (9)$$

$$\overleftarrow{h_t} = \mathcal{H}(\mathbf{W}_{x\overleftarrow{h}}x_t + \mathbf{W}_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}) \quad (10)$$

$$y_t = \mathbf{W}_{\vec{h}y}\vec{h}_t + \mathbf{W}_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y \quad (11)$$

where $\mathcal{H}$ is the composite function. The BLSTM allows to exhibit long range context dependencies and takes advantage from the two directions structure. The output vector y is processed by evaluating simultaneously the two directions hidden sequences by computing the composite function H in the forward and backward directions.

## 5.2 Unified Multi-View Interaction Model

Although sequence based approaches to satisfaction prediction are an effective way of capturing user interactions, we hypothesize that better, richer representation of user activity can be obtained by incorporating other interaction signals in the model architecture. The traditionally used static features and implicit signals provide a different view of the user interactions. Our primary contribution here is a novel neural architecture that is designed to jointly leverage sequence information with such static implicit feedback signals to predict search satisfaction.

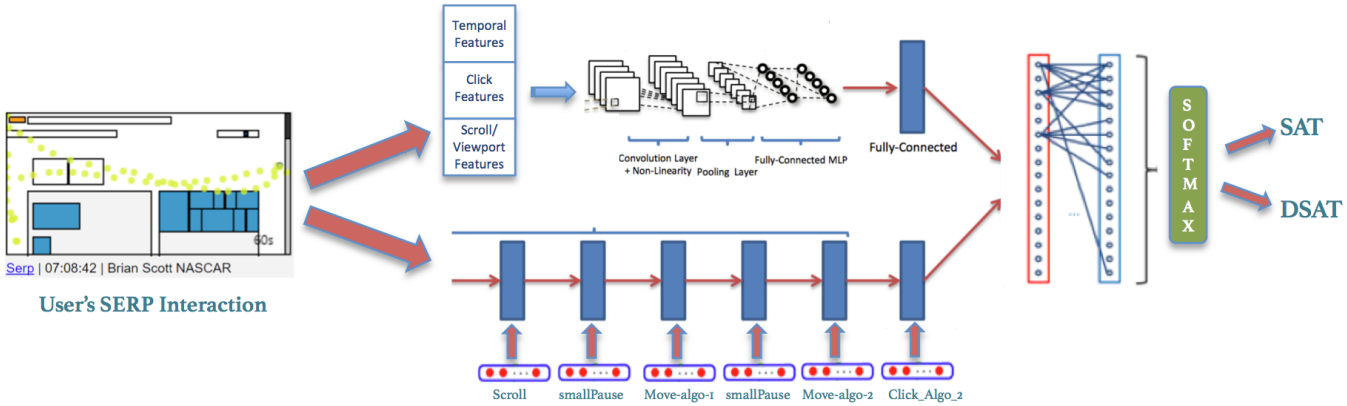


**Figure 2: Neural architecture of the proposed deep Unified Multi-view CNN-LSTM model.**

| Feature Set | Feature List |
|---|---|
| Temporal Signals | Page dwell time |
| | Reading time per pixel |
| | Viewport time per instance |
| | Time to first pointer event |
| | Time to first scroll event |
| Click based Signals | Total click count |
| | Algo click count |
| | Answer click count |
| Scroll & Pointer Signals | Total scroll count |
| | Pointer horizontal distance |
| | Pointer vertical distance |
| | Pointer event count |
| | Scroll Up count |
| | Scroll down count |
| | Viewport direction changes |

**Table 3: List of auxiliary signals used as side information.**

To illustrate, consider the example of a user action sequence: Pause – Scroll – Click. While sequence information is informative, aggregate metrics such as dwell times etc provide useful cooked information and are helpful in capturing domain information about user behavior with SERP.

*5.2.1 Auxiliary Signals.* A number of different interaction behaviors have been taken into consideration in the prediction of search user satisfactions including both coarse-grained features (e.g. clickthrough based features [11]) and fine-grained ones (e.g. cursor position and scrolling speed [10]). We use a number of such traditionally used signals as auxiliary side-information which provides an alternative view of user interaction. We categorize these signals into three groups: (i) Temporal signals, (ii) Click based signals and (iii) Scroll & pointer signals. Table 3 presents the different types of signals captured under each of these groups which provide us an alternative view of the user interaction. We next describe our model which jointly encodes these auxiliary features with the sequential action-LSTM model.

*5.2.2 Unified Multi-View Interaction Model.* The auxiliary signals described above provide us with an alternate view of user interaction. We use these auxiliary signals to enrich our sequential model to create a unified multi-view model of user interactions. We propose a coupled architecture composed of deep convolutional network and dense layers for modelling auxiliary features and couple it with the action-LSTM architecture described before. Its main building blocks are (i) action-LSTM which use the action sequences and (ii) the auxiliary feature module based on convolutaional neural networks (ConvNets), both of which work in parallel mapping details of user interactions to their distributional vectors which are then used to predict user satisfaction for each query.

The architecture of our ConvNet for mapping implicit signals to features is mainly inspired by the various CNN architectures used for performing different classification tasks. However, different from previous work the goal of our distributional auxiliary signals model is to learn good intermediate representations of such signals, which are then coupled together with the output representation of the sequential action-LSTM model and used for satisfaction prediction. The input to the ConvNet module are the three set of implicit feedback signals (as shown in Table 3) that are processed by intermediate convolutional layers. The aim of the convolutional layers is to extract patterns, i.e., discriminative signal sequences found within the input signals that are common throughout the training instances.

More specifically, the convolution operator operates on sliding windows of signals, and the convolutions in deeper layers are defined in a similar way. Suppose we have a discrete input function $g(x) \in [1, l] \rightarrow R$ and a discrete kernel function $f(x) \in [1, k] \rightarrow R$. The convolution $h(y) \in [1, \lfloor (l-k)/d \rfloor + 1] \rightarrow R$ between f(x) and g(x) with stride $d$ is defined as:

$$h(y) = \sum_{x=1}^{k} f(x) \cdot g(y \cdot d - x + c) \tag{12}$$

where c = k - d + 1 is an offset constant. The module is parameterized by a set of such kernel functions $f_{ij}(x)(i = 1, 2, ..., m$ and $j = 1, 2, ..., n)$ which we call weights, on a set of inputs $g_i(x)$ and outputs $h_j(y)$. The output from the convolutional layer (passed through the activation function) are then passed to the pooling layer, whose goal is to aggregate the information from the previous layer. Given a discrete input function $g(x) \in [1, l] \rightarrow R$, we employ a 1-D spatial max-pooling function $h(y) \in [1, \lfloor (l-k)/d \rfloor + 1] \rightarrow R$ of g(x) defined as:

$$h(y) = max_{x=1}^{k} g(y \cdot d - x + c) \tag{13}$$

where c = k - d + 1 is an offset constant. To allow the network learn non-linear decision boundaries, each convolutional layer is typically followed by a non-linear activation function applied element-wise to the output of the preceding layer. The non-linearity used in our

model is the rectifier or thresholding function

$$h(x) = max\{0, x\} \tag{14}$$

which makes our convolutional layers similar to rectified linear units (ReLUs) [36] acting as a non-linear feature extractor. Finally, we combine the ConvNet feature extractor with the output of the action-LSTM and pass it through two dense layers. and a softmax layer at the end.

**Interaction Layers:**
Our model includes an additional hidden layer right before the softmax layer (described next) to allow for modelling interactions between the components of the intermediate representation, i.e., the different views of user interactions. The hidden layer computes the following transformation: $\alpha(w_h \cdot x + b)$ where $w_h$ is the weight vector of the hidden layer and $\alpha()$ is the ReLU non-linearity function.

**Softmax Layer:**
The output of the penultimate convolutional and pooling layers is flattened to a dense vector x, which is passed to a fully connected softmax layer. It computes the probability distribution over the labels:

$$p(y = j|x) = \frac{e^{x^T \theta_j}}{\sum_{k=1}^{K} e^{x^T \theta_k}} \tag{15}$$

where $\theta_k$ is a weight vector of the k-th class. x can be thought of as a final abstract representation of the input example obtained by a series of transformations from the input layer through a series of convolutional and pooling operations.

### 5.3 Training

The parallel multi-view CNN-LSTM model is trained to minimize the RMSE error on satisfaction prediction accuracies. We use the ADAM optimization algorithm for training [23], with a batch size of 64. The learning rate is initially chosen as 0.01, and dropped to 0.003 in the middle of training before convergence. We useed the standard default values for other parameters of the optimizer: $\beta_1$ = 0.9, $\beta_2$ = 0.999, $\epsilon = 10^{-8}$. While neural networks have a large capacity to learn complex decision functions they tend to easily overfit especially on small and medium sized datasets. To mitigate the overfitting issue we insert 2 dropout modules in between the fully-connected layers to regularize. They have dropout probability of 0.2. Dropout prevents feature co-adaptation by setting to zero (dropping out) a portion of hidden units during the forward phase when computing the activations at the softmax output layer and also acts as an approximate model averaging [39].

## 6 FUNCTIONAL COMPOSITION FOR TASK SATISFACTION

Given details of user interactions at the query level and the corresponding query level satisfaction prediction architecture proposed in the previous section, our overall goal is to make task satisfaction predictions. In this section, we enumerate different ways of using query satisfaction predictions to make task level satisfaction predictions. Specifically, given a sequence of queries $Q = q_1, q_2, ..., q_t$ belonging to a search task $t \in T$, where $T$ is the set of all tasks, the Multi-view CNN-LSTM architecture provides us with estimates of query level satisfaction $Y_{q_i} = \varphi_q(q_i, a_{q_i})$ where $Y_{q_i} \in \{0, 1\}$ is the query level satisfaction estimate, $a_q$ is the set of action sequence observed for the search impression for query $q$ and $\varphi_q$ is the query level satisfaction prediction function. Our goal is to make task level satisfaction prediction:

$$y_t = F\Big(\big\{q_1, \varphi_q(q_1)\big\}, \big\{q_2, \varphi_q(q_2)\big\}, \big\{q_3, \varphi_q(q_3)\big\}, ..., \big\{q_i, \varphi_q(q_t)\big\}\Big) \tag{16}$$

where **F**: $\{q_1, \varphi_q(q_1)\} \rightarrow Y_t \in 0, 1$ represents the functional transformation which maps query-satisfaction estimate tuple $\{q_i, \varphi_q(q_i)\}$ to a task satisfaction label. Based on known insights on task satisfaction, we present a number of different functional compositions techniques at two levels of abstract: (i) query level aggregation and (ii) subtask level aggregation.

### 6.1 Query level composition

To make task level satisfaction predictions, we aggregate satisfaction signals at the query level. We consider four distinct functional forms of aggregation, ranging from extremely strict to lenient evaluation of task satisfaction.

(1) **Maximum**: This functional composition method assumes that the user is satisfied in completing their task if they are satisfied in any of the queries they issued while completing the task. Specifically:

$$y_t = max\Big(\varphi_q(q_1), \varphi_q(q_2), \varphi_q(q_3), ..., \varphi_q(q_t)\Big) \tag{17}$$

where $\varphi_q(q_i)$ gives the query level satisfaction estimate based on the Multi-View CNN-LSTM architecture. It is to be noted that such a functional composition is the most lenient way of evaluating search engine performance.

(2) **Average**: This functional composition techniques considers equal contribution from each query in predicting task satisfaction. Specifically, $y_t = \frac{\sum_{i=1}^{|Q_t|} \varphi_q(q_i)}{|Q_t|}$ where $|Q_t|$ is the number of queries associated with the task $t$.

(3) **Differential Weighting:** Often users reformulate their information needs and issue a series of queries as they complete their task. We hypothesize that queries towards the end of the task are more important than the ones at the start, based on which we over emphasize queries towards the end of the task when considering their contribution towards task satisfaction. Specifically:

$$y_t = \frac{\sum_{i=1}^{|Q_t|} w_i \varphi_q(q_i)}{|Q_t|} \tag{18}$$

where $w_i$ is the weight associated with query $q_i$.

(4) **Minimum**: This functional composition assumes that a user is satisfied is completing their task if they are satisfied in each of the queries they issued to accomplish the task. Specifically:

$$y_t = min\Big(\varphi_q(q_1), \varphi_q(q_2), \varphi_q(q_3), ..., \varphi_q(q_t)\Big) \tag{19}$$

Such an computation of task satisfaction is the most strict estimate of task satisfaction since if any SERP rendered for

query is unsatisfying to the user, the whole task is rendered unsatisfying.

## 6.2 Subtask based composition

Often search tasks involve many distinct, but related aspects which warrant the need for issuing different sets of queries over time in order to fulfill the multi-aspect information needs. A complex search task could be broken down into smaller multi-step or multi-aspect sub-tasks that represent atomic informational needs, for which it is trivial for users to issue satisfying queries. We hypothesize that task-level satisfaction could be estimated based on user's satisfaction levels when attempting different subtasks. An ideal task completion engine would help the user satisfactorily accomplish each of the associated subtasks. We utilize this insight to estimate task satisfaction from the associated subtask satisfaction estimates.

Given a task $t$ composed of $|S_t|$ subtasks, we consider a nested functional composition of satisfaction estimates at two levels: (i) aggregating query satisfaction estimates to compute subtask satisfaction and (ii) aggregating subtask satisfaction estimates to compute task satisfaction. Specifically,

$$y_t = f\Big(g\Big(\{\varphi_q(q_i)\}\forall q_i \in S_1\Big), g\Big(\{\varphi_q(q_i)\}\forall q_i \in S_2\Big), \ldots, g\Big(\{\varphi_q(q_i)\}\forall q_i \in S_t\Big)\Big)$$

where $S_i$ represents the subtask $j$ and $S_t$ represents the total number of subtasks in the task $t$. The functions f(.) and g(.) could be either of the four query level aggregate functions defined before. While there exist automated subtask extraction approaches [32], for the scope of this work, we assume access to subtask demarcation information obtained via crowdsourced labeling.

# 7 EXPERIMENTAL EVALUATION

In this section, we demonstrate how our satisfaction prediction models perform for predicting both query and task level satisfaction. We conduct a number of experiments using crowdsourced judgments as well as real world search engine traffic. We make use of labels obtained via crowdsourced judgments studies as ground truth labels for all evaluations considered; however, we leverage large scale pseudo-labeled data with weak supervision signals to train our deep models.

## 7.1 Dataset

Our data consists of a random sample of user sessions from a major US commercial search engine engine during a week in June 2016. We randomly sampled user sessions with substantial user activity, and included all queries, search result page impressions on all results on the search result page from that user in the timeframe. Additionally, detailed user activity on the result page was logged for model development. In total, our sample contained No of sessions over 14670 search sessions, resulting in about 148561 search queries.

*7.1.1 Large Scale Pseudo-Labelled Data.* While we collect crowdsourced labels for creating ground truth labels, owing to the limited scale of experimentation possible with crowd-sourced judgments as well as the differences in opinion of crowdsourced judges and actual users, we may have insufficient data and labels to reliably train deep parameter-rich models. To resolve this problem, we build a pseudo-labeled dataset comprised of the entire large-scale query log described in Section 7.1. To assign pseudo satisfaction labels to search interactions, we assume that a click followed by a query reformulation is a dissatisfied click, while a click with a dwell time of $\geq 30$ seconds not followed by a query reformulation is a satisfied click. Post-click query reformulation is considered a strong DSAT predictor and has been used as a predictor of search satisfaction in previous work [15, 22]. To identify query reformulations we use a method similar to that described in Boldi *et al.*[4], where features of query similarity (e.g. edit distance, word overlap, etc.) and time between queries are used to identify query reformulations.

## 7.2 Collecting Task SAT Judgements

Crowdsourced judgments have commonly been used to obtain labeled data [43, 44]. To gauge user satisfaction at both query level and task level, we collect judgement labels at both levels. For each search impression as well as the overall task, we obtained human labeled judgments on whether the user interaction was satisfying (labelled SAT) or not (labelled DSAT). The labelling was conducted using an in-house microtasking platform that outsources crowd work to vendors, similar to CrowdFlower, and provides access to judges who regularly perform relevance judgment tasks. Workers were under NDA and all data containing personal identifiable information (PII), such as names, phone numbers, addresses, or social security numbers, were removed.

Detailed guidelines were issued to the judges to describe the task and a number of examples were shown defining what constitutes a query, a subtask and a task and explaining how to judge for query as well as task level satisfaction. To ensure the quality of the judging results, we apply a series of quality control methods. One of the methods is creating 'gold hits' that you already know the answer of, then measure the judges by comparing how far off their answers are from the gold hits answers. We also measure the quality of the judgments with the amount of consensus reached which required overlap on the hits.

The data presented to the judges come from previously annotated data where another group of judges defined the task boundaries within a session. In other words, each session was divided into one of more coherent tasks. A sequence of queries are considered part of a coherent task if they collectively try to achieve a certain goal. The output of the task boundary annotation is given to our group of judges where each is represented as a series of queries along with the corresponding user interaction information. In order to provide relevant information to the judges, we provided a detailed summary of user interaction with the SERP. The judges were provided a link to the SERP shown to the user alongside details like number of clicks, time spent on the SERP and scroll information. Additionally, for all the clicked documents, we provided URL level details which included the exact URL, the position on the SERP where it was shown and the total dwell time on each URL. Each judge was asked to consider the user interaction summary and provide labels for query and task satisfaction.

We randomly sampled over 2100 user tasks and over 450 judges provided judgments for about 6820 search impressions, resulting in over 20460 judgments. Among the first two judgments collected for

| Method Type | Method | Accuracy | Precision | Recall | FMeasure | Log-Loss |
|---|---|---|---|---|---|---|
| Feature based baselines | Baseline 1(Clicks+Dwell Time) | 0.561 | **0.86** | 0.58 | 0.6927 | 13.88 |
| | Baseline 2 (Click based actions) | 0.593 | 0.78 | 0.61 | 0.6846 | 13.67 |
| | Baseline 3 (Mouse Movement) | 0.606 | 0.72 | 0.66 | 0.6886 | 13.32 |
| | Baseline 4 (Scroll & Viewport)) | 0.586 | 0.71 | 0.67 | 0.6894 | 13.73 |
| | Baseline 5 (Reading Pattern Signals) | 0.596 | 0.72 | 0.69 | 0.7046 | 13.61 |
| Sequential baselines | Generative Probabilistic | 0.631 | 0.81 | 0.67 | 0.7333 | 13.04 |
| | CRF-Actions | 0.593 | 0.77 | 0.6 | 0.6744 | 14.74 |
| | CRF-Queries | 0.582 | 0.75 | 0.62 | 0.6788 | 14.89 |
| Proposed/Variants | SimpleRNN | 0.654 | 0.72 | 0.85 | 0.7796 | 11.36 |
| | action-Embedding + LSTM | 0.668 | 0.71 | 0.88 | 0.7859 | 11.08 |
| | action-Embedding + Bi-LSTM | **0.677**$^{*\&}$ | 0.73 | **0.89**$^{*\&}$ | **0.8020**$^{*\&}$ | **10.98**$^{*\&}$ |

**Table 4:** Query level SAT prediction. $^{*}$ and & indicate statistical significant ($p \leq 0.05$) using paired t-tests compared to the best performing feature based baseline and the best performing sequential baseline respectively.

| Method | Accuracy | Precision | Recall | FMeasure | Log-Loss |
|---|---|---|---|---|---|
| CRF-Actions | 0.593 | 0.77 | 0.6 | 0.6744 | 14.74 |
| CRF-Actions + All Signals | 0.603 | 0.78 | 0.61 | 0.6848 | 14.29 |
| Generative Probabilistic | 0.631 | 0.81 | 0.67 | 0.7333 | 13.04 |
| Generative Probabilistic + All Signals | 0.651 | 0.823 | 0.681 | 0.744 | 12.97 |
| action-Embedding+ LSTM | 0.677 | 0.73 | 0.89 | 0.8020 | 10.98 |
| action-Embedding+ LSTM + Click based Signals | 0.678 | 0.744 | 0.825 | 0.783 | 11.11 |
| action-Embedding+ LSTM + Temporal Signals | 0.699 | 0.714 | **0.954**$^{*\&}$ | **0.817**$^{*\&}$ | 10.36 |
| action-Embedding+ LSTM + Scroll/Viewport Signals | 0.689 | **0.728**$^{*\&}$ | 0.89 | 0.801 | 10.72 |
| Unified Multi-View Model (action-LSTM + All Signals) | **0.703**$^{*\&}$ | 0.717 | 0.944 | 0.815 | **10.25**$^{*\&}$ |

**Table 5:** Evaluating the unified model for Query SAT prediction. $^{*}$ and & indicate statistical significant ($p \leq 0.05$) using paired t-tests compared to the CRF all signals and Generative Probabilistic - All Signals baselines respectively.

each query, the judges agreed on the label 74% of the time. We measured inter-rater agreement using Fleiss' Kappa [6], which allows for any number of raters and for different raters rating different items. This makes it an appropriate measure of inter-rater agreement in our study since different judges provided labels for different items. A kappa value of 0 implies that any rater agreement is due to chance, whereas a kappa value of 1 implies perfect agreement. In our data, $\kappa = 0.64$, which, according to Landis and Locke [28], represents substantial agreement.

## 7.3 Baselines

We consider a number of baselines from recent published literature, including both non-neural and neural models, as well as non-sequential and sequence based models.

- **Baseline 1 (click with dwell time):** Spending a minimum amount of time on a webpage is known as a long dwell click and has been shown to be correlated with satisfaction [22]. In this study, we set t = 30 seconds.
- **Baseline 2 (click based actions):** This baseline is based on predicting satisfaction based on clickthrough based features [10].
- **Baseline 3 (Mouse movement):** This baseline is based on recent work aimed at predicting satisfaction using mouse movement patterns [29].
- **Baseline 4 (Scroll & Viewport):** This baseline is based on the recently proposed scrolling and viewport features [26, 44]
- **Baseline 5 (Reading pattern signals):** This baseline is based on the reading pattern signals from Kiseleva *et al.*[24]

Additionally, we consider a number of sequence based models to compare the performance of the proposed approach.

- **Generative Probabilistic Model:[13]** A semi-supervised generative model wherein every action sequence is generated using a probability distribution specified by a 2-component mixture model.
- **CRF Models:** Conditional random field models are popularly used for many different sequence labeling tasks. We consider two variants of CRF models:
  - action-CRF: this CRF makes use of only the action information for constructing CRF features.
  - query-CRF Model: in addition to action co-occurrence features, this CRF model takes into account query level features during training.

We also consider variants of the proposed model: (i) simple RNNs, (ii) action-embedding LSTM and (iii) action-Embedding Bi-LSTM.

## 7.4 Query Level SAT Prediction

As our first experiment, we consider predicting user satisfaction for each search impression. We compare the proposed sequential action embedding + LSTM model with traditionally used features as well as other popular feature based and sequential models. For each query, we extract the set of features needed by the different baselines as well as the detailed user interaction action sequence and consider the judgment labels obtained from the crowdsourced study as the ground truth. We randomly split the data into training and test set in 60/40 ratio. We use the pseudo-labelled data described in Section 7.1.1 to pre-train the neural models.

Table 4 presents the prediction results comparing the proposed approach with established baselines. We observe that sequence based baselines perform better than feature based baselines in general, with the generative probabilistic baseline performing particularly better with over 7% improvement in accuracy scores. A satisfying click, i.e., click followed by a long dwell time, has traditionally been used to gauge user satisfaction. We re-confirm such

| Functional Composition Type | Method | Accuracy | Precision | Recall | FMeasure | Log-Loss |
|---|---|---|---|---|---|---|
| Maximum | CRF | 0.639 | 0.914 | 0.629 | 0.7454 | 12.46 |
| | Generative Probabilistic | 0.811 | **0.917** | 0.852 | 0.883 | 6.49 |
| | action-Embedding Bi-LSTM | 0.823 | 0.841 | 0.974 | 0.902 | 6.08 |
| | Unified Multi-View | **0.8309**$^{*\&}$ | 0.838 | **0.988**$^{*\&}$ | **0.907**$^{*\&}$ | **5.83**$^{*\&}$ |
| Minimum | CRF | 0.5259 | 0.785 | 0.589 | 0.679 | 16.37 |
| | Generative Probabilistic | **0.826** | 0.838 | **0.983** | **0.904** | **6.003** |
| | action-Embedding Bi-LSTM | 0.618 | **0.94**$^{*\&}$ | 0.356 | 0.517 | 19.32 |
| | Unified Multi-View | 0.5952 | 0.882 | 0.597 | 0.712 | 13.98 |
| Average | CRF | 0.57 | 0.906 | 0.544 | 0.6807 | 14.82 |
| | Generative Probabilistic | 0.797 | **0.918** | 0.832 | **0.873** | 6.99 |
| | action-Embedding Bi-LSTM | 0.6809 | 0.847 | 0.756 | 0.799 | 11. |
| | Unified Multi-View | **0.801**$^{*\&}$ | 0.842 | **0.895**$^{*\&}$ | 0.868 | **6.89**$^{*\&}$ |
| Differentially Weighted | CRF | 0.632 | 0.913 | 0.62 | 0.739 | 12.7 |
| | Generative Probabilistic | 0.814 | **0.917** | 0.855 | 0.885 | 6.41 |
| | action-Embedding Bi-LSTM | 0.714 | 0.853 | 0.781 | 0.815 | 8.21 |
| | Unified Multi-View | **0.824**$^{*\&}$ | 0.849 | **0.929**$^{*\&}$ | **0.887**$^{*}$ | **6.84**$^{*\&}$ |
| Subtask (Max-Average) | CRF | 0.591 | **0.926** | 0.553 | 0.6924 | 13.66 |
| | Generative Probabilistic | 0.761 | 0.901 | 0.812 | 0.8541 | 8.89 |
| | action-Embedding Bi-LSTM | 0.70 | 0.83 | 0.79 | 0.82 | 10.36 |
| | Unified Multi-View | **0.77**$^{*\&}$ | 0.84 | **0.89**$^{*\&}$ | **0.86**$^{*}$ | **8.14**$^{*\&}$ |
| Subtask (Average-Max) | CRF | 0.621 | 0.904 | 0.632 | 0.7439 | 12.89 |
| | Generative Probabilistic | 0.814 | **0.921** | 0.841 | 0.8791 | 6.41 |
| | action-Embedding Bi-LSTM | 0.79 | 0.85 | 0.92 | 0.88 | 7.56 |
| | Unified Multi-View | **0.838**$^{*\&}$ | 0.84 | **0.98**$^{*\&}$ | **0.91**$^{*\&}$ | **6.08**$^{*\&}$ |

**Table 6:** **Task level SAT prediction. $^{*}$ and & indicate statistical significant ($p \leq 0.05$) using paired t-tests compared to the CRF and Generative Probabilistic baselines respectively.**

known insights since we observe that Click+DwellTime obtain the best precision; however this method misses out on capturing various other satisfactory interactions, as is evident from their low recall scores. Further, we observe that mouse movement information (baseline 3) in general is more predictive than just click based features.

Overall, we observe that the proposed deep sequential model and its variants outperform all baselines considered in predicting user satisfaction and register an improvement in over 11% over the worst performing baseline and ~5% over the best performing generative sequence modelling approach. Among the variants considered, the simple RNN model is outperformed by the more sophisticated LSTM models which confirms known benefits offered by LSTMs over RNNs. The bidirectional version of the proposed model outperforms the LSTM model on all metrics, which confirms our hypothesis that including future action signal information helps in modeling user interaction better. Indeed, since most satisfaction detection and evaluation is performed post-hoc, and historic data logs entire user interactions, future actions signal information is readily available and should be used in modeling user interactions. The proposed deep sequential models perform significantly better in terms of recall, with obtaining 20% improvement over the best performing baselines. This strongly suggests that the rich user interaction signals used by our deep sequence models are perhaps able to capture and detect user satisfaction in non-click scenarios, and abandonment cases.

## 7.5 Unified View for QSAT

We next evaluate the benefit of unifying the different interactions signals, both static features and interaction sequences. We investigate how adding different sets of features to the sequential model help in better predicting user satisfaction. Table 5 presents the results on query level satisfaction prediction comparing the proposed Unified Multi-View CNN+LSTM model with the best performing baselines.

| Method | Accuracy | Precision | Recall | FMeasure | Log-Loss |
|---|---|---|---|---|---|
| CRF | 0.639 | **0.914** | 0.629 | 0.7454 | 12.46 |
| Generative Probabilistic | 0.826 | 0.838 | 0.983 | 0.9045 | 6.003 |
| action-Embedding Bi-LSTM | 0.823 | 0.841 | 0.974 | 0.9022 | 6.08 |
| Unified Multi-View | **0.843**$^{*\&}$ | 0.851 | **0.991**$^{*\&}$ | **0.9156**$^{*\&}$ | **5.62** $^{*\&}$ |

**Table 7:** **Comparing the performance of different task SAT prediction approaches across all functional compositional techniques. $^{*}$ and & indicate statistical significant ($p \leq 0.05$) using paired t-tests compared to the CRF and Generative Probabilistic baselines correspondingly.**

We observe that adding the other view of user interaction data always helps in improving prediction performance across all methods. Adding click based signal information to the interaction sequence information improves SAT precision (at the cost of recall), which is consistent with what was observed before. Adding temporal signals give a significantly improved performance in terms of recall, with over 27% improvement in detecting satisfaction cases which may have otherwise been missed by baseline approaches. Indeed, temporal signals and detailed user interactions go well beyond shallow methods which assume a very restrictive view of user satisfaction. Further, the unified multi-view model achieves the best accuracy in predicting user satisfaction with over 5% improvement in accuracy, 26% improvement in recall and 7% improvement in F-score. These results strongly demonstrate the benefits offered by the enriched unified multi-view models by leveraging not only the interaction sequence information, but also other static implicit signals.

## 7.6 Task SAT Prediction

One major motivation for the current work is to leverage user interaction signals to predict task level satisfaction of users. To this end, we consider the problem of task satisfaction prediction and compare how the different compositional functions perform in predicting task level satisfaction. Since we collected task satisfaction judgements alongside query level satisfaction judgements, we make use of these task level judgements as ground truth information.

Before diving deep into different compositional functions, we first look at how the proposed models perform on the task satisfaction problem. As shown in Table 7, we observe that the proposed deep sequential model performs better than the best performing baselines in predicting task satisfaction across all five metrics. Moreover, the unified multi-view model performs better than the deep sequential model, which demonstrates that the combined information from interaction sequences and other auxiliary implicit feedback signals are not only good for query level satisfaction prediction, but also work best at measuring task satisfaction.

We additionally analyze how the different functional composition techniques fare. Table 6 presents the task satisfaction prediction results wherein we compare the proposed models with best performing baselines across the different functional composition techniques. We considered five different functional composition techniques for aggregating query level satisfaction estimates to compute task satisfaction. We observe that the most lenient aggregating technique (*Maximum*) consistently achieves higher accuracy than the most strict satisfaction criterion (*Minimum*). We observe that the differential weighting scheme performs better than the average function, which hints at the fact that not all queries contribute the same towards a task. Finally, considering subtasks information in the intermediary stage between query and task level abstractions helps in better predicting task satisfaction.

## 8 CONCLUSION & FUTURE WORK

We considered a holistic view of user interaction and presented deep sequential models for predicting user satisfaction at various levels of abstraction. While most exiting approaches focus on query satisfaction or task satisfaction for simple atomic tasks, we go beyond such atomic tasks and consider the problem of predicting user's satisfaction when engaged in complex search tasks composed of many different queries and subtasks. The proposed unified multi-view model and the functional composition approach performs better than a number of established baselines. We hope that the findings of this work would inspire future research in developing sophisticated techniques for quantifying the importance of different queries and subtasks in any given complex task. Further, task satisfaction prediction could inspire research in developing retrieval algorithms optimized for task completion. Finally, we contend that the promising results demonstrated by the unified multi-view approach would help in improving satisfaction prediction and good abandonment detection on mobile devices.

## REFERENCES


[1] Eugene Agichtein, White, Dumais, and Bennet. Search, interrupted: understanding and predicting search task continuation. In *SIGIR 2012*.
[2] Ahmed, White, Pantel, Dumais, and Wang. Supporting complex search tasks. In *Proceedings of the ACM CIKM 2014*.
[3] Ioannis Arapakis, Mounia Lalmas, and George Valkanas. Understanding within-content engagement through pattern analysis of mouse gestures. In *CIKM 2014*.
[4] Paolo Boldi, Francesco Bonchi, Carlos, Debora, Aristides, and Sebastiano. The query-flow graph: model and applications. In *CIKM 2008*.
[5] Henry A Feild, James Allan, and Rosie Jones. Predicting searcher frustration. In *SIGIR 2010*.
[6] Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin* (1971).
[7] Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais, and Thomas White. 2005. Evaluating implicit measures to improve web search. *ACM TOIS* (2005).
[8] Qi Guo and Agichtein. Beyond dwell time: estimating document relevance from cursor movements and other post-click searcher behavior. In *WWW 2012*.
[9] Qi Guo and Eugene Agichtein. Exploring mouse movements for inferring query intent. In *SIGIR 2008*.
[10] Qi Guo, Dmitry Lagun, and Eugene Agichtein. Predicting web search success with fine-grained interaction data. In *CIKM 2012*.
[11] Qi Guo, Ryen W White, Susan T Dumais, Jue Wang, and Blake Anderson. 2010. Predicting query performance using query, result, and user interaction features. In *Adaptivity, Personalization and Fusion of Heterogeneous Information*.
[12] Martin T Hagan and Mohammad B Menhaj. 1994. Training feedforward networks with the Marquardt algorithm. *IEEE transactions on Neural Networks* (1994).
[13] Ahmed Hassan. A semi-supervised approach to modeling web search satisfaction. In *SIGIR 2012*.
[14] Ahmed Hassan, Rosie Jones, and Kristina Lisa Klinkner. Beyond DCG: user behavior as a predictor of a successful search. In *WSDM 2010*.
[15] Ahmed Hassan, Xiaolin Shi, Nick Craswell, and Bill Ramsey. Beyond clicks: query reformulation as a predictor of search satisfaction. In *CIKM 2013*.
[16] Jeff Huang, Ryen W White, Georg Buscher, and Kuansan Wang. Improving searcher models using mouse cursor activity. In *SIGIR 2012*.
[17] Jeff Huang, Ryen W White, and Susan Dumais. No clicks, no problem: using cursor movements to understand and improve search. In *CHI 2011*.
[18] Jiepu Jiang, Ahmed Hassan Awadallah, Xiaolin Shi, and Ryen W White. Understanding and predicting graded search satisfaction. In *WSDM 2015*.
[19] Rosie Jones and Kristina Lisa Klinkner. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *CIKM 2008*.
[20] Diane Kelly. 2009. Methods for evaluating interactive information retrieval systems with users. *Foundations and Trends in Information Retrieval* (2009).
[21] Youngho Kim, Ahmed Hassan, White, and Zitouni. Comparing client and server dwell time estimates for click-level satisfaction prediction. In *SIGIR 2014*.
[22] Youngho Kim, Ahmed Hassan, Ryen W White, and Imed Zitouni. Modeling dwell time to predict click-level satisfaction. In *WSDM 2014*.
[23] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[24] Julia Kiseleva, Kyle Williams, Hassan Awadallah, Crook, Zitouni, and Anastasakos. Predicting user satisfaction with intelligent assistants. In *SIGIR 2016*.
[25] Alexander Kotov, Paul N Bennett, Ryen W White, Susan T Dumais, and Jaime Teevan. Modeling and analysis of cross-session search tasks. In *SIGIR 2011*.
[26] Dmitry Lagun, Ageev, Qi Guo, and Agichtein. Discovering common motifs in cursor movement data for improving web search. In *WSDM 2014*.
[27] Dmitry Lagun, Chih-Hung Hsieh, Webster, and Navalpakkam. Towards better measurement of attention and satisfaction in mobile search. In *SIGIR 2014*.
[28] J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics* (1977).
[29] Yiqun Liu, Ye Chen, J Tang, J Sun, M Zhang, S Ma, and Zhu. Different users, different opinions: Predicting search satisfaction with mouse movement information. In *SIGIR 2015*.
[30] Yiqun Liu, Wang, Ke Zhou, Nie, Zhang, and Ma. From skimming to reading: A two-stage examination model for web search. In *CIKM 2014*.
[31] Claudio Lucchese, Salvatore Orlando, Raffaele Perego, Fabrizio Silvestri, and Gabriele Tolomei. Identifying task-based sessions in search engine query logs. In *WSDM 2011*.
[32] Rishabh Mehrotra, Prasanta Bhattacharya, and Emine Yilmaz. 2016. Deconstructing Complex Tasks. In *Proceedings of NAACL*.
[33] Rishabh Mehrotra and Emine Yilmaz. 2017. Extracting Hierarchies of Search Tasks & Subtasks via a Bayesian Nonparametric Approach. In *Proceedings of SIGIR 2017*. ACM, 285–294.
[34] Rishabh Mehrotra, Imed Zitouni, Ahmed Hassan, Ahmed Kholy, and Madian Khabsa. 2017. User Interaction Sequences for Search Satisfaction Prediction. In *Proceedings SIGIR 2017*. ACM.
[35] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
[36] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML 2010*.
[37] Kerry Rodden, Xin Fu, Anne Aula, and Ian Spiro. Eye-mouse coordination patterns on web search results pages. In *CHI 2008*.
[38] Denis Savenkov, Dmitry Lagun, and Liu. Search engine switching detection based on user personal preferences and behavior patterns. In *SIGIR 2013*.
[39] Srivastava. 2014. Dropout: a simple way to prevent neural networks from overfitting. *JMLR* (2014).
[40] Louise T Su. 1992. Evaluation measures for interactive information retrieval. *Information Processing & Management* (1992).
[41] Hongning Wang, Yang Song, Ming-Wei Chang, Xiaodong He, Ryen W White, and Wei Chu. Learning to extract cross-session search tasks. In *WWW 2013*.
[42] Ryen W White and Susan T Dumais. Characterizing and predicting search engine switching behavior. In *CIKM 2009*.
[43] Ryen W White, Matthew Richardson, and Wen-tau Yih. Questions vs. queries in informational search tasks. In *WWW 2015*.
[44] Kyle Williams, Julia Kiseleva, Aidan C Crook, Zitouni, Awadallah, and Khabsa. Detecting good abandonment in mobile search. In *WWW 2016*.