Task Duration Estimation

Ryen W. White
Microsoft Research
Redmond, WA 98052
ryenw@microsoft.com

Ahmed Hassan Awadallah
Microsoft Research
Redmond, WA 98052
hassanam@microsoft.com

ABSTRACT

Estimating how long a task will take to complete (i.e., the task duration) is important for many applications, including calendaring and project management. Population-scale calendar data contains distributional information about time allocated by individuals for tasks that may be useful to build computational models for task duration estimation. This study analyzes anonymized large-scale calendar appointment data from hundreds of thousands of individuals and millions of tasks to understand expected task durations and the longitudinal evolution in these durations. Machine-learned models are trained using the appointment data to estimate task duration. Study findings show that task attributes, including content (anonymized appointment subjects), context, and history, are correlated with time allocated for tasks. We also show that machine-learned models can be trained to estimate task duration, with multiclass classification accuracies of almost 80%. The findings have implications for understanding time estimation in populations, and in the design of support in digital assistants and calendaring applications to find time for tasks and to help people, especially those who are new to a task, block sufficient time for task completion.

CCS CONCEPTS

• **Human-centered computing**; • **Computing methodologies** → *Machine learning*; • **Information systems** → *Data mining*;

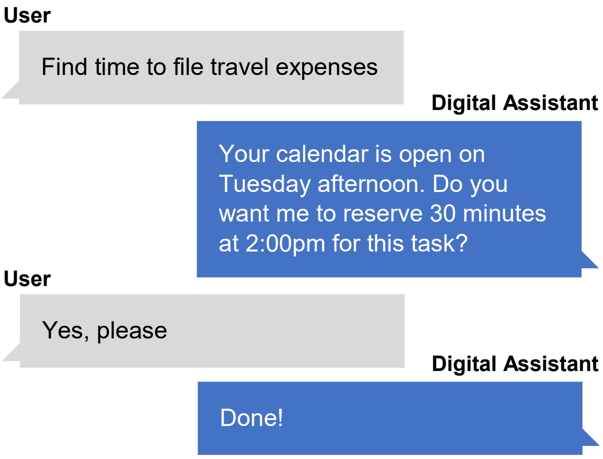
KEYWORDS

Time estimation; Task duration; Intelligent scheduling

ACM Reference Format:
Ryen W. White and Ahmed Hassan Awadallah. 2019. Task Duration Estimation. In *The Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19), February 11–15, 2019, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3289600.3290997

1 INTRODUCTION

Tasks (i.e., pieces of work to be performed) permeate all aspects of our lives. Determining how much time to allocate for tasks is an important aspect of effective time management that is often performed manually. Time estimates impact how much time people set aside for tasks during planning and when they perform tasks based on estimates of how long they need. The activity of time estimation can be challenging for users, especially for new tasks, when people may lack the experience or expertise to make accurate time estimates [17, 41]. Time estimates can also be affected by known biases, such as optimism and overconfidence [22, 30].



User

Find time to file travel expenses

Digital Assistant

Your calendar is open on Tuesday afternoon. Do you want me to reserve 30 minutes at 2:00pm for this task?

User

Yes, please

Digital Assistant

Done!

Figure 1: Example of dialog between a user and a digital assistant that is equipped with the ability to accurately estimate task duration and schedule time for completion.

Despite the importance of task duration estimation in our daily lives (every meeting or appointment we schedule requires us to make this decision), there has been little research on helping users perform duration estimation. Previous work has focused on support for scheduling appointments of known duration given a user's calendar and (optionally) others' constraints [6, 13]. Work on time estimation has largely focused on how people perceive task duration [5] or have only discussed using distributional data about how much time others have taken for the same task, but have not implemented any technical solutions [9]. The availability of anonymized population-scale calendar data creates opportunities to understand and learn task duration, both at scale and individually over time.

In this paper, we study methods for estimating task duration, i.e., we train machine-learned models that, given a task, provide an estimate of how long that task will take to complete. Making this determination automatically, even if it is only used to suggest a task duration as part of task or appointment creation, may ease some burden on users. Figure 1 presents an example of how this technology could be applied in a dialog with a digital assistant. In the example, the assistant uses the task duration estimate to suggest an appointment length for the task (30 minutes to file travel expenses in this case) and find a suitable timeslot based on available time on the user's calendar (Tuesday at 2:00pm in this case). For users attempting a task for the first time, suggestions such as these—generated based on distributional (historic) data

WSDM '19, February 11–15, 2019, Melbourne, VIC, Australia
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-5940-5/19/02…$15.00
https://doi.org/10.1145/3289600.3290997

and reflecting the amount of time others have allocated for similar tasks—may be especially helpful in estimating the time required for task completion.

We make the following contributions with this research:

- Introduce *task duration estimation* as a new data mining and machine-learning challenge, with a range of applications.
- Analyze millions of anonymized calendar events (tasks), each with a user-defined task duration. We find correlations between task attributes / task recurrence over time and task duration.
- Train machine-learned models to accurately predict task duration from task attributes. We also experiment with different model architectures (logistic regression and neural networks) and different features (content, context, history).
- Present implications for digital assistants, to-do applications, and calendaring systems from being able to estimate task duration, as well as future directions for research in this area.

The remainder of this paper is structured as follows. Section 2 describes related work in areas such as time management and time estimation. Section 3 describes the anonymized appointment data used in our study. In Section 4, we present an analysis of task duration, focused on the relationship between task attributes (including task recurrence) and task duration. Section 5 presents methods for automatically estimating task duration and the results of their evaluation are presented in Section 6. We discuss our findings and their implications in Section 7 and conclude in Section 8.

2 RELATED WORK

Tasks have received significant attention in research areas such as information seeking and retrieval [10, 28]. Focusing specifically on the challenge of estimating task duration, there are several key areas of related previous work. In this section, we target two areas in particular: personal time management and time estimation.

2.1 Time Management

Research has shown that those who perceive themselves as good time managers are most accurate at the estimating duration of a future task [14]. Of those who do not perceive themselves as good time managers, some people grossly overestimate, and many people underestimate, the time required. Prior work on personal time management has focused on best practices to help people manage their time more effectively [1] and on developing tools to better support this activity [3, 31]. Intelligent scheduling systems can help individuals [18, 34] and groups [6, 13] find time for tasks and meetings, coordinating schedules between attendees as needed in the group setting. The focus of these systems is on finding calendar slots that satisfy constraints about meeting times and locations. Digital assistants such as Amazon Alexa, Google Assistant, and Microsoft Cortana provide timers to help people track short time durations and enable users to create reminders to remember to perform future tasks [15], even if the specific task timeframe is imprecise [36]. Support for micro-tasking [11] helps people utilize small amounts of time, even if just a few minutes, to tackle quick to-dos or to make progress on larger (macro) tasks.

2.2 Time Estimation

Time estimation has been well studied [9, 14], including biases such as anchoring that may impact the accuracy of time estimates [20, 23]. The planning fallacy [8, 22], where people underestimate the time taken to complete their own tasks (days/weeks) [7] is often based on "singular information" (not distributional information as we have access to in this study) related to the specific task and an optimism bias (or wishful thinking [33] or overconfidence [30]), and irrespective of how long previous similar tasks have taken to perform [7, 25, 26]. When tasks are easy (minutes), people have been shown to overestimate duration [4, 14]. Time perceptions have also been studied before the task (expected), during the task (prospective), and after the task (retrospective) [5, 38], including the effects of experience [41], expertise [17], and motivation [40] on task duration estimates. The important role of attentional demand in time estimation has also been demonstrated [44].

Distributional information (e.g., base-rate data on previous task performance) is important in forecasting task duration [9, 22]. People often ignore or lack access to distributional data [21], leading to the planning fallacy described earlier. Other research has shown that people may use distributional information in time estimation, but inaccurately recall it when making time predictions [39]. People may focus too much on the task at hand and too little on the time they spend on previous similar tasks [7]. That said, focusing too much on prior tasks may also introduce biases that are difficult to ignore [24, 37, 42]. A significant strength of this study is that we have data on actual time allocated to tasks by large populations of users. We use that data to train models to more objectively generate time estimates that are unaffected by shortcomings in human memory and time perceptions.

2.3 Contributions Over Previous Work

There are several key differences between our research and previous work. First, support for time management has mostly focused on ways to find time or make effective use of time, whereas we focus on helping to determine how much time is needed. Second, while work on human time estimation has targeted biases and subjective perceptions in task duration, we focus on objective estimates of how much time will be needed to complete a task. We use the creation of a calendar appointment as the provision of a task, with a description and task duration estimate. Studying these events over hundreds of thousands of individuals and millions of anonymized calendar appointments enables population-scale analysis and machine learning of task duration. Data were collected from a natural setting, from users of a popular digital assistant deciding how they would like to allocate their own time for work and life activities. Finally, prior work on automatically estimating duration has only discussed the use of distributional (historic) data about time spent or allocated by others on similar tasks. In contrast, we train machine-learned models from anonymized large-scale data; those models utilize a broad range of content, context, and history signals, in addition to historic time distributions, for automatic duration estimation.

3 TASKS DATASET

The study uses anonymized calendar data collected from consenting users of the Cortana digital assistant over a period of 18 months,

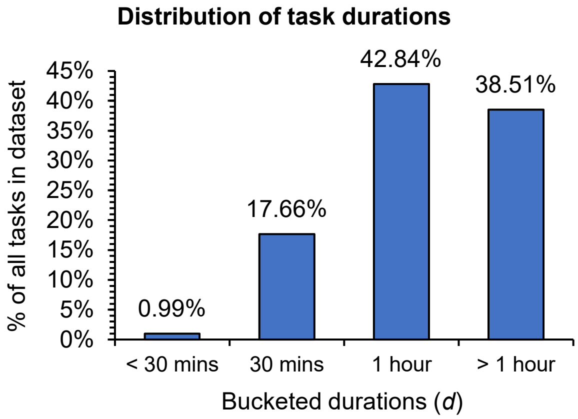


Figure 2: Distribution of discretized task durations (d) for tasks in our tasks dataset (D) (n=2,748,285).

from December 2016 to May 2018 (inclusive). A random sample of millions of calendar events were available in the data, including calendar appointments added by users explicitly (where they are the only attendee), events added automatically by the digital assistant or other applications, and meetings involving the user and other individuals. All data (including the text of the appointment subjects) were anonymized as part of an initial data processing step. For consistency, all appointments were filtered to US English using a language classifier applied to the subject during initial data processing.[1] A separate classifier was required because language and locale information were unavailable in the original appointment dataset. To preserve privacy, text tokens in appointment subjects were replaced with one-way hashes near the start of the data processing pipeline. Prior to anonymizing the text, we extracted attributes such as whether an entity was present and frequency counts for each part of speech (POS) tag.[2]

Appointments where the user is the only attendee and the corresponding time allocated for the task were used as the primary data source for analysis and learning. To obtain these events, the appointments dataset was filtered to calendar events that were intentionally placed on the calendar by users and met all of the following criteria: (1) organized by one user; (2) accepted by the user (not canceled, declined, or tentative); (3) the user is the only attendee (not meetings with others); (4) not flagged as an all-day appointment unless the user manually blocks that time;[3] (5) not created as a recurring appointment (although the same appointment could still be scheduled multiple times (discussed more in Section 4.2)); (6) not holidays, birthdays, travel, hotel reservations, out of office, or any automatic inferences (e.g., commute) identified using whitelist lookup – all of which were reminders rather than time reservations; and (7) had a time span of 86,400 seconds (one day) or less. Appointments running longer than one day are typically blocked in one-day increments, which we deemed was not sufficiently granular for our initial analysis of task duration.

Table 1: Pearson correlations between task attribute and task duration (d), filtered to attributes with abs(r) ≥ 0.05.

Attribute	*r*
Mean duration (per user-task)	+0.41487
Median duration (per user-task)	+0.41056
Mean duration (per task)	+0.35767
Median duration (per task)	+0.35035
Mean duration (per user)	+0.32666
Median duration (per user)	+0.30167
Has location †	+0.23894
Std deviation duration (per user-task)	+0.14959
Std deviation duration (per user)	+0.14557
Maximum token length †	+0.05715
Has country †	+0.05117
Has address †	+0.05024
Minimum token length †	-0.05258
Start minute	-0.05719
Has phone number †	-0.06061
Fraction text stop words (a, the, etc.) †	-0.07673
Total number of stop words in text †	-0.08378
Number of unique stop words in text †	-0.08741
Task popularity (across all users)	-0.13104
Number of action verbs †	-0.14705

† Non-(time/duration/history) attributes are based on task description

We believe that meeting all of these constraints qualifies each of the filtered appointments as a user task, with the appointment subject as the task description. Examples of popular task descriptions that emerged from these pre-processing steps included "file expenses," "call insurance broker," and "read chapter 2." These appointments made up the dataset D, contain 2.75M task descriptions and associated durations from 596K users. The average number of tokens in the task descriptions was 4.64 (standard deviation=5.11, median=4). The average time span from first to last task per user in D was 22.03 days (standard deviation=135.69 days, median=0 days (most users have only one task in D)). Analyzing all appointments in D reveals an overall average task duration (d) of 7,321 seconds (2.03 hours) and a standard deviation of 11,380 seconds (3.16 hours). The median d was 3,600 seconds (one hour) and 60.5% of appointments lasted between 30 and 60 minutes, inclusive. Figure 2 presents the distribution of d, discretized into four buckets (classes). These specific buckets align with our intended application of duration estimation (intelligent scheduling, illustrated in Figure 1 and discussed more later in the paper). Figure 2 shows that there is a fair spread in amount of time allocated for tasks, especially in the 30 minute-plus time range.

4 ANALYZING TASK DURATION

We now present some analysis of the task durations in D. Specifically, we analyze two aspects of d: (1) the relationship between task attributes and d, and (2) the relationship between task recurrence over time and d.

4.1 Task Attributes

To understand the relationship between attributes of the task itself and the allocated task duration, we computed the Pearson correlation (r). Several attributes of the task could be computed based

[1] https://www.nuget.org/packages/NTextCat

[2] Part-of-speech tags were added using RDRPOSTagger [32].

[3] Many appointments with an "all day" flag appeared to be reminders to do a task at some time on that day and did not contain realistic user-defined task durations.

on the text content of the task description (appointment subject), the context (time and location), and history (prior individual and population task duration statistics). Table 1 reports the results of this analysis for attributes with an absolute correlation value of 0.05 or above (to focus on the most salient correlations). Findings show that historic task attributes, generated from the time before the start of the current task, including the time that the current user spent on this same task historically, are most correlated with d. Task effects are strong, and time on task ("per user-task," specific to the current user and task and "per task," specific only to the current task, but across many users) is more strongly correlated with d than the time that a user historically spends on all of their tasks ("per user"). Physical location attributes, such as whether the task (appointment) has a location, country, or address are also positively correlated with d ($0.05 < r < 0.24$). The frequent need to allow time for travel may mean that these tasks take longer. In contrast, in the last eight rows of Table 1, there are task attributes that are associated with shorter task duration. Attributes that are negatively correlated with d often contain telephone numbers, are frequently occurring, contain verbs, and use more basic language (shorter tokens, more stop words). Common sense suggests that tasks with these attributes are clearer and/or simpler, and hence might require less time to complete. These attributes and others are used as features in the task duration estimation models and associated experiments presented later in the paper.

4.2 Task Recurrence

The importance of history as a task attribute prompted us to analyze changes in duration estimates for the same task repeated multiple times (i.e., task occurrence (i) of $i \geq 2$). To measure this in retrospective analysis, appointments in D created by the same user with the same subject were regarded as recurrences of the same task. This heuristic is used rather than the recurring appointment flag from many calendaring systems because appointments that are flagged as recurring also have the same duration; offering no opportunity to study how people change their duration estimates.

In D, there were 767K appointments (28% of total) with $i \geq 2$. The average duration was computed for tasks at each i, in this case in the range [1,10] for tasks occurring ≥ 10 times (n=9,956). Figure 3 shows the average task duration and the variance in the estimation (standard error) as we sweep i from 1-10. To avoid skew from a single user with many tasks, we first average for each ⟨user, i⟩ pair and again for each i. To make the comparison with the overall average more interpretable, we normalize task duration using the z-score and include a line-of-best-fit using a linear regression.

There are two noteworthy trends in Figure 3. First, durations for tasks that recur are generally higher than the overall average (i.e., all z-scores exceed 0). Recurring tasks have a higher average d (7,496 seconds) than the overall average d across all tasks (7,321 seconds). Second, task duration tends to increase from i=1 (first occurrence of the task) to i=10 (tenth occurrence) (R^2=0.7052, p-val < 0.05). There are many explanations for this increase, including planning fallacies [22] where people may be updating their task duration estimates over time following prior underestimates. The increase may also reveal characteristics of the tasks where people frequently block time on their calendars, i.e., not easy tasks, for which they often overestimate duration [4, 14], as evidenced by the small number of tasks with d < 30 minutes in D (1% per Figure 2).

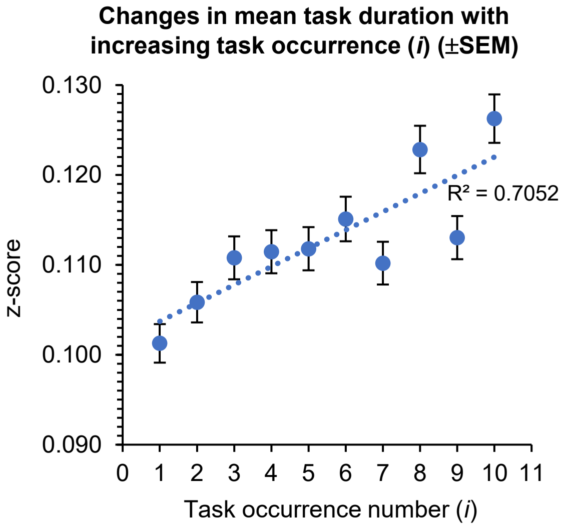


Figure 3: Average duration of appointments per the occurrence number (i). Error bars denote the standard error of the mean (±SEM). Dotted line is line of best fit, showing an upward trend in task duration alongside occurrence number.

The clear trends and patterns from this analysis are promising for automatically estimating duration from task attributes (including task recurrence). We investigate that in detail in the next section.

5 ESTIMATING TASK DURATION

In this section, we discuss methods for training machine-learned models to accurately estimate task duration from task attributes. We begin by formally defining our estimation task and we then introduce the task attributes we can use for estimation. Finally, we describe two methods for training a task duration estimator: one using logistic regression and the other using neural networks.

5.1 Problem Definition

As discussed earlier, we study the problem of task duration estimation and we hypothesize that we can assist users by estimating the time needed to complete tasks based on several task attributes. Given a task, our objective is to estimate how much time would be needed to accomplish the task. Note that this problem could be posed either as a regression problem (estimating time needed for each task given task attributes) or a classification problem (assign a discrete duration category to each task). We chose to formulate the problem as a classification problem because of (1) 87.4% of durations in our dataset are multiples of 30 minutes and (2) task duration estimation is intended to enable applications in digital assistants and calendaring applications where an agent assists the user by automatically blocking time on his or her calendar to perform a pending task. If we were to pose the problem as a regression problem, we would still need to postprocess the model output to assign it to discrete categories. As such, to better align the problem definition with our main application (scheduling, as in Figure 1), we decided to treat duration estimation as a multiclass classification problem. We use the four classes in Figure 2 as our target labels: < *30 minutes*, *30 minutes*, *1 hour*, and > *1 hour*. There are several task attributes available for featurization. We now describe the features used to represent tasks and our duration estimation models.

5.2 Task Features

Our data contains three main categories of attributes about each task that we can leverage for the estimation task. The attributes are listed in Table 2 and we describe each category of attributes below:

Content attributes focus on the description of the task entered by the user when the task was created. As discussed in Section 3, text tokens were replaced with hashes. The hashed version of the text description was available as an attribute for the machine learning models. Additionally, several other content features were computed prior to hashing all tokens. That included counting the number of tokens, the number of stop words, etc. Additionally, we flagged entities such as locations and organizations, as well as first names and phone numbers. We also flagged action verbs and depth words (that capture thoroughness such as "deep," "detailed," "careful," "thorough," "overview," "lightweight," "light," "end to end," "e2e") and recorded their occurrence frequency. We also counted each POS tag. Finally, we generated a 300-dimensional vector representing a sentence embedding for the text attribute using the model in [19]. We hypothesize that content information derived from the task description could help us estimate task duration. Content information is by far the most important source of information since other sources such as context and history information may not be always available (e.g., new users with little or no history).

Context attributes characterize contextual information related to the task. Context information could be related to the time or the location of the task. To represent task time, we use several attributes such as day of the week, time of the day, etc. the task was scheduled to start. To characterize the location, we identify whether the task description contains information about a location, address, or a mention of a country name. Note that the information we have about the location is rather limited and we try to infer location information from the text rather than using the actual user location when the task was created or accomplished. We leave collecting and leveraging such information to future work. We hypothesize that time and location information may be useful for describing the context at which the task is performed and hence can benefit task duration estimation.

History information for each task includes information before the start of the task. We describe the popularity and duration of tasks using the number of tasks, average duration, median duration, and standard deviation duration. We compute these statistics per task, per user, and per user-task pair. We hypothesize that historic information could help in estimating task duration, and that making it available to the model could allow the model to learn specific patterns from task history to improve its estimates.

5.3 Duration Estimation Models

The attributes described above could be used as features in a traditional classification model, such as logistic regression. All features in Table 2 are numerical except for the text description of the task. Numerical features can be fed directly to the model post normalization. We generate *n*-grams (up to 3-grams) from the text description of the task and use *n*-gram TF-IDF values as features.

Table 2: Task attributes as features for duration estimation.

Name	Description
Content Features	
Text	Hashed tokens in the task description
NumTokens	Number of tokens (words)
MaxTokenLength	Num. chars. of the longest token
MinTokenLength	Num. chars. of the shortest token
AvgTokenLength	Average num. characters per token
NumStopwords	Number of stop words
FractionStopwords	Fraction of text that are stop words
NumEntities	Number of entities
NumNames	Number of person names
HasPhoneNumber	Whether text has a phone number
NumActionWords	Number of action words
NumDepthWords	Number of depth words
NumPOSTags	Frequency count for each POS tag
SentVec	A 300-d sentence embedding vector
Context Features	
StartDayOfWeek	Day of the week
StartMonth	Month of the year
StartDayOfYear	Day of the year
StartHour	Start hour
StartMinute	Start minute
HasLocation	Whether description contains location
HassAddress	Whether description contains address
HasCountry	Whether description contains country
History Features	
Num_Task	Task frequency (per task)
AvgDur_Task	Mean duration (per task)
MedDur_Task	Median duration (per task)
StdDur_Task	Std. deviation duration (per task)
Num_User	Task frequency (per user)
AvgDur_User	Mean duration (per user)
MedDur_User	Median duration (per user)
StdDur_User	Std. deviation duration (per user)
Num_UserTask	Task frequency (per user-task) †
AvgDur_UserTask	Mean duration (per user-task)
MedDur_UserTask	Median duration (per user-task)
StdDur_UserTask	Std. deviation duration (per user-task)

† Note this is equivalent to task occurrence index (*i*) (Section 4.2)

As we discussed earlier, we believe that content information is extremely important for the challenge of task duration estimation. This could be attributed, in part, to the fact that other sources of information may not be available in a cold-start scenario where history information is unavailable. Motivated by the recent advances in applying neural network methods to natural language understanding tasks, we propose an approach for modeling the task description with a recurrent neural network architecture.

Given an appointment subject S with a list of words $w_i, i \in 1..n$, we aim to embed S into a fixed size representation vector. For each word w_i in the subject, we first transform them into dense vectors through a word embedding matrix $W \in R^{dim \times |V|}$. Here, $|V|$ is the size of vocabulary, and dim is the dimensionality of the word embedding. Typically, we could use pre-trained word embeddings such as word2vec [29]. However, in our case this is not possible since in this study we can only operate on the anonymized (token hashed) version of the text descriptions for tasks. As such, we train our own word embedding vectors using a skip-gram hierarchical SoftMax model [29] using Gensim [35].

The embedding vector e_w for a word w is obtained by multiplying the one-hot vector representation of the word w with the embedding matrix. After placing words with their embedding vectors, we apply a bi-directional RNN with GRU cells to the anonymized appointment subject S. The bi-directional RNN contains two RNNs, a forward RNN that scans the text from the beginning to the end and a backward RNN that scans the text in the opposite direction. We obtain the hidden state h_i for each word w_i in the appointment subject S by concatenating the forward hidden state and the backward hidden state. Bi-directional RNNs scan the subject from both sides, and hence allow each word hidden state to encode information about the text before and after the corresponding word.

To generate a single vector representing the whole subject S, we could use aggregation functions such as max-pooling or averaging over the hidden state of each word in S. Alternatively, we could allow the model to learn the optimal way to combine the word representations into a sentence representation while taking account of words or phrases that are more important than others for estimating the task duration. To accomplish this, we leverage the attention mechanism [2] and use the weighted average of the hidden states to represent S. The attention mechanism takes all hidden states H as input, and outputs a weight vector α as:

$$\alpha = softmax(v_s \tanh(W_s H^T)) \tag{1}$$

where $H = [h_1, h_2, \ldots, h_n]$, and v_s and W_s are learned attention parameters. The representation of $S(e_s)$ is obtained as follows:

$$e_s = H\alpha^T \tag{2}$$

Finally, we pass the sentence representation to a SoftMax (a multiclass logistic regression) to generate the final estimation. In the cases where we would like to include additional numerical features (other than text), we concatenate the feature vector with the sentence representation and pass them to the SoftMax classifier.

6 EXPERIMENTS

In this section, we describe the experimental setup and the results of the estimation experiments. We also include a detailed analysis of the performance of the feature categories and learning methods.

6.1 Content-Only Models

We start by describing the experiments for task duration estimation using only content features. This setting is particularly important because it does not require any history information and hence can be applied to all users, regardless of their activity level. We compare the results of four different machine-learned models:

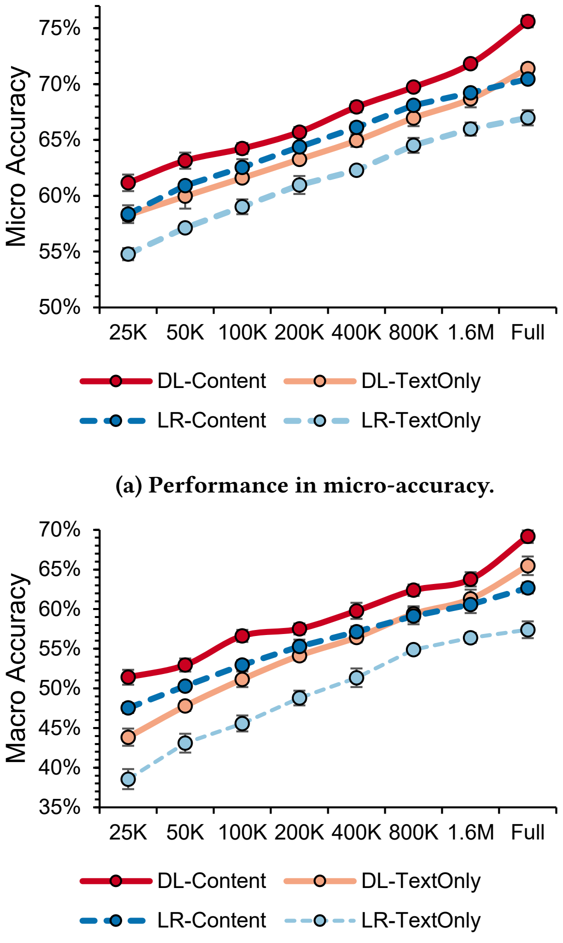
- **LR-TextOnly:** A logistic regression model using n-gram (up to 3 grams) TF-IDF features from the hashed text description.
- **LR-Content:** The same as *LR-TextOnly* but also uses the rest of the precomputed content features described in Table 2.
- **DL-TextOnly:** The bidirectional RNN model with the GRU cells and attention mechanism described in Section 5.3. The input to the model is the text description only.
- **DL-Content:** The same as *DL-TextOnly* but a numerical feature vector representing the precomputed content feature is concatenated with the sentence representation before passing it to the SoftMax classifier.

In addition to comparing the four models, we also vary the training set size from 25K instances to 2.75M instances and observe the performance of each model. We split the dataset such that we have a validation set of 20K instances and a test set of 20K instances. We used the validation set to tune all hyperparameters (L1 and L2 weights for logistic regression and batch size, learning rate, dropout rate, and GRU hidden unit size for deep learning).

The results are shown in Figure 4. The top portion of the figure shows the performance in micro-accuracy and the lower portion shows the performance in macro-accuracy. We report both metrics for completeness. Micro-accuracy treats all test cases equally (and may be preferable if there is class imbalance, as Figure 2 suggests), whereas macro-accuracy treats each of the four classes equally. Additionally, we report the per class F1 measure, as well as the average and weighted average F1 for all classes, in Table 3. *LR-TextOnly* performs the worst, achieving a micro-accuracy and a macro-accuracy of 67% and 57.4% respectively when the full dataset is used. Adding other precomputed content features (see Table 2) to the logistic regression model (*LR-Content*) considerably improves the performance, increasing the micro-accuracy and macro-accuracy to 70.5% and 62.7% respectively when the full dataset is used. For both logistic regression models, we also observe that the performance improves as the training dataset size increases, but the growth slows significantly as we add more data beyond 800K training examples.

Now we turn our attention to the deep learning models described in Section 5.3. Using the text only (*DL-TextOnly*) outperforms *LR-TextOnly* at all training dataset sizes. It does not perform as well as *LR-Content* when the training dataset size is small (100K or less) but it performs better as the training dataset size increases, achieving a micro-accuracy and a macro-accuracy of 71.4% and 65.5% respectively when the full dataset is used. Adding the other content features to the deep learning model has a similar effect to adding them to the logistic regression model and *DL-Content* outperforms all other variants for all training dataset sizes and achieves a micro-accuracy and a macro-accuracy of 75.6% and 69.2% respectively, when the full dataset is used. Additionally, the deep learning models (*DL-TextOnly*, *DL-Content*) appear to benefit considerably from the increase in the training set size and the performance continues to increase as more data is added. In summary, all content-based models appear to be able to estimate task duration with reasonable accuracy, with the deep learning models achieving the best performance. This is an important result, since it shows that we could reliably estimate task duration relying only on the text description of the task and in the absence of any context or history signals.

To better understand the type of errors the model makes, we examine all incorrect predictions made by *DL-Content* when trained on the full training set. For each class, we compute the percentage of instances of this class incorrectly assigned to each other class. The results are shown in Table 4. For each row in the table, we



(a) Performance in micro-accuracy.

(b) Performance in macro-accuracy.

Figure 4: Performance in terms of micro-accuracy (top) and macro-accuracy (bottom) of various content-based models with various training set sizes. *-*TextOnly* models use the normalized text only, while *-*Content* models use additional content-based features covered in Section 5. Recall that "Full" has 2.75M tasks. Error bars denote ±SEM.

show the percentage of incorrectly classified instances for this class with respect to all other classes. The table shows that most mistakes assign instances to adjacent classes, suggesting that errors are typically not too egregious. For example, 66.6% of errors for the < *30 minutes* class were assigned to the *30 minutes* class and 66.1% of the errors for the *30 minutes* class are assigned to the *1 hour* class. The errors for the *1 hour* class are almost evenly split between the *30 minutes* and the > *1 hour* class. Similarly, 76.2% of errors in the > *1 hour* class were assigned to the *1 hour* class instead. The fact that most errors happen between adjacent classes further supports our choice of treating task duration estimation as a multiclass classification problem as opposed to a regression problem.

Table 3: Per class F1 measure for different models using the full dataset for training. The average and weighted average (wAvg) F1 measure for each model across all classes are reported on the right of the table.

Model	Class				Avg	wAvg
	< 30 mins	30 mins	1 hour	> 1 hour		
LR-TextOnly	49.3%	51.7%	69.3%	72.4%	60.7%	67.1%
LR-Content	68.4%	54.9%	71.4%	75.0%	67.4%	69.8%
DL-TextOnly	69.9%	56.4%	72.1%	75.7%	68.5%	70.6%
DL-Content	68.8%	66.4%	75.1%	78.6%	72.2%	74.8%

Table 4: Distribution of errors across classes. Each row represents all prediction errors for instances belonging to one class. The percentages show how often each other class is incorrectly predicted for the given (true) class.

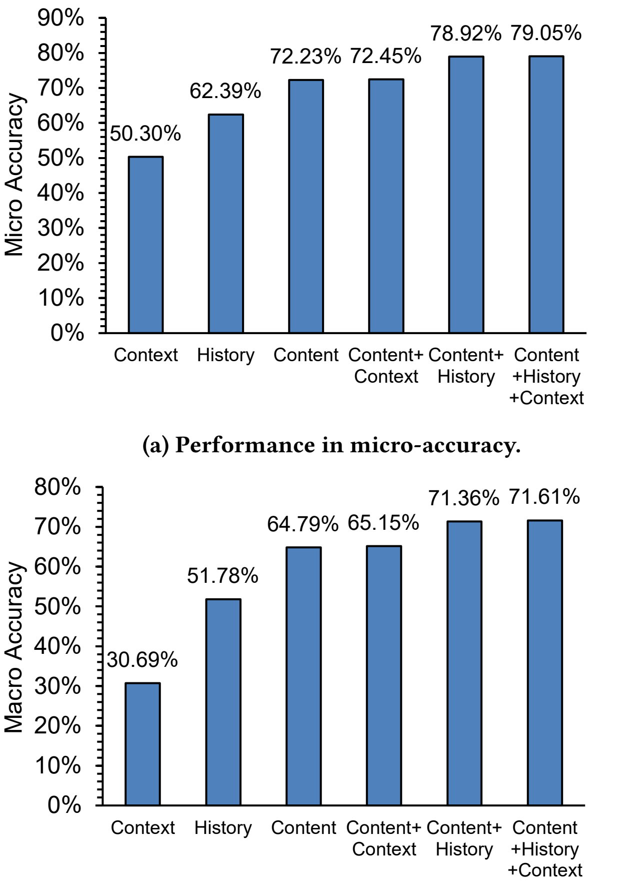
		Predicted			
		< 30 mins	30 mins	1 hour	> 1 hour
Truth	< 30 mins	0.0%	66.6%	16.7%	16.7%
	30 mins	3.6%	0.0%	66.1%	30.3%
	1 hour	1.1%	44.1%	0.0%	54.8%
	> 1 hour	0.7%	23.1%	76.2%	0.0%

The focus thus far was on four-class classification. Alternatives might make this simpler and result in improved classification performance, e.g., accuracy at three-class duration estimation $d \in \{\leq$ *30 minutes*, *1 hour*, > *1 hour* $\}$ using the *DL-Content* model and the full training dataset was 75.1% and 77.1% for micro- and macro-accuracies respectively (vs. 75.6% and 69.2% for four-class estimation). Note that micro-accuracy did not improve due to the small size of the < *30 minutes* class (1% per Figure 2). Future work could also subdivide the > *1 hour* class into smaller classes to provide more duration alternatives (e.g., two hours, half-day, full-day).

6.2 Effect of History and Context

So far, we have focused on leveraging content only features for task duration estimation. However, for a subset of users we may have additional information such as task context and history information (see Table 2 for details). We ran several experiments to estimate the impact of adding these features to the best performing content-based model (*DL-Content*). We concatenate the numerical feature vectors generated from context and history information to the representation of the task description generated by the encoding described in Section 5.3 and pass the vector to the SoftMax (multiclass logistic regression) classifier. We also experimented with using the task context and history features alone by passing them to a multiclass logistic regression classifier directly.

Since history information is not available for all users. We use a subset of users that had three or more tasks in our dataset. This excluded almost half of the dataset leaving us with 1.4M instances. We generated a new temporal train/test split with 85% of the data used for training and 15% held out for testing. The temporal split ensures that the classifier only has access to information from the past and is evaluated on future tasks. This mimics the scenario where such a model could be used in practice.



(a) Performance in micro-accuracy.

(b) Performance in macro-accuracy.

Figure 5: Performance in micro-accuracy (top) and macro-accuracy (bottom) of various models leveraging *Content* information, *Context* information, *History* information, and various combinations of the three categories.

The results of this experiment are shown in Figure 5. To measure statistical significance, we used two-tailed *t*-tests with bootstrap sampling. The first three bars in the figure compare content, history, and context information. Context information seems to perform poorly, while history information seems to yield better performance. Compared to context and history information, content information clearly results in the best performance (p-val $\ll$ 0.05). Adding context information to the content model results in limited but still statistically-significant accuracy gains (p-val > 0.05). Conversely, adding history information to content models results in a much larger improvement (6.7 and 6.6 points in micro- and macro-accuracy respectively) (p-val $\ll$ 0.05). Adding both history and context information to the content models does not seem to yield meaningful improvements over adding only history information (p-val > 0.05). This suggests that history information, when available, could yield significant gains in the accuracy of task duration estimation models. Conversely, context information does not appear to add value in addition to content and history information. Note that our context information represented both the time and the location of performing the task. The latter was rather limited since location context had to be obtained by detecting location mentions (e.g., city, state) in the task description. Additional work may be needed to find better features to represent the task context.

7 DISCUSSION AND IMPLICATIONS

The evidence from this study suggests that by using distributional signals and user histories, computer systems can learn to estimate task duration, bringing us closer to methods to work alongside humans to help them better manage their time. Promisingly, our findings align with prior work on time estimation, especially on planning fallacies [22] and the impact of previous task experience [41]. We extend those studies, which are often small scale and do not address the challenge of learning to estimate task duration.

The results show that our classification accuracy is well above the marginal baseline (42.8%) and that deep learning improves over logistic regression. For completeness, we report both macro- and micro-averages; but we also need to explore alternative metrics [16]. Classification performance is strong with content only and improves as we add user histories and more training data. Histories may not be available for many tasks, and the need to double training data to significantly improve accuracy is not scalable. More sophisticated learning algorithms and richer features are required to realize additional gains. For example, prior work found a relationship between lead time and time estimation accuracy [27, 43]. We experimented with using lead time and other context signals and saw only limited accuracy gains; although as noted earlier, our representations of context were also quite limited. If duration estimators were applied for other applications, such as estimating the duration of tasks inferred from email communications (e.g., commitments or requests [12]) then additional information about the sender, recipient, the task, etc. may be available from email metadata and content, and could be used for duration estimation.

In our analysis, we assumed that appointment duration is the actual time spent on task. We need to validate that assumption and explore ways to obtain data at scale on actual time spent. Some of the biases outlined earlier in the paper (e.g., [22]) may affect our duration estimations, especially since updates to appointment duration after the appointment are likely to occur infrequently. Figure 3 suggests that the appointment durations are refined over time and tend to increase across multiple occurrences. Recurring appointments may contain more reliable time estimates to train and test duration estimation models. However, these appointments may also be homogeneous and less representative of the broad range of tasks for which people allocate their time.

Applications of task duration estimation in digital assistants and calendaring applications could be useful (e.g., for the scenario in Figure 1), but also requires further study and refinement. Beyond single-person appointments, there is an opportunity to expand these methods to include tasks with multiple people, where there are additional social and work-task factors that affect task duration. Combining task duration estimators with other methods, say, to predict when tasks are most likely to be performed [15] could get us closer to time management solutions that allocate the required duration at the best time (not just any suitable block of free time).

8 CONCLUSIONS

Task duration estimation is an important aspect of time management. Access to anonymized large-scale data on time allocated to tasks let us train models to accurately estimate duration. We showed the impact of different signals (content, context, history) on model accuracy. Duration estimation is a new machine learning challenge and our model performance (~80% accuracy) is promising. This has several implications for scheduling systems and digital assistants, e.g., suggest time slots with enough time to complete the current task. Future work will run user studies to understand user preferences, explore alternative learning strategies, create shareable datasets to drive further research, and train models using data about actual time on task in addition to times reserved on calendars.

REFERENCES

[1] David Allen. 2001. *Getting things done: the art of stress free productivity.* New York: Viking.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[3] Ann E Blandford and Thomas RG Green. 2001. Group and individual time management tools: what you get is not what you need. *Personal and Ubiquitous Computing* 5, 4 (2001), 213–230.

[4] Marilyn G Boltz, Cara Kupperman, and Jessica Dunne. 1998. The role of learning in remembered duration. *Memory and Cognition* 26, 5 (1998), 903–921.

[5] Scott W Brown. 1985. Time perception and attention: the effects of prospective versus retrospective paradigms and task demands on perceived duration. *Perception and Psychophysics* 38, 2 (1985), 115–124.

[6] Mike Brzozowski, Kendra Carattini, Scott R Klemmer, Patrick Mihelich, Jiang Hu, and Andrew Y Ng. 2006. groupTime: preference based group scheduling. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems.* ACM, 1047–1056.

[7] Roger Buehler, Dale Griffin, and Michael Ross. 1994. Exploring the "planning fallacy": Why people underestimate their task completion times. *Journal of Personality and Social Psychology* 67, 3 (1994), 366.

[8] Roger Buehler, Dale Griffin, and Michael Ross. 1995. It's about time: Optimistic predictions in work and love. *European Review of Social Psychology* 6, 1 (1995), 1–32.

[9] Christopher DB Burt and Simon Kemp. 1994. Construction of activity duration and time management potential. *Applied Cognitive Psychology* 8, 2 (1994), 155–168.

[10] Katriina Byström and Kalervo Järvelin. 1995. Task complexity affects information seeking and use. *Information Processing and Management* 31, 2 (1995), 191–213.

[11] Justin Cheng, Jaime Teevan, Shamsi T Iqbal, and Michael S Bernstein. 2015. Break it down: A comparison of macro-and microtasks. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems.* ACM, 4061–4064.

[12] Simon Corston-Oliver, Eric Ringger, Michael Gamon, and Richard Campbell. 2004. Task-focused summarization of email. In *Proceedings of the ACL-04 Workshop: Text Summarization Branches Out.* 43–50.

[13] Justin Cranshaw, Emad Elwany, Todd Newman, Rafal Kocielnik, Bowen Yu, Sandeep Soni, Jaime Teevan, and Andrés Monroy-Hernández. 2017. Calendar.help: designing a workflow-based scheduling agent with humans in the loop. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems.* ACM, 2382–2393.

[14] Jan A Francis-Smythe and Ivan T Robertson. 1999. On the relationship between time management and time estimation. *British Journal of Psychology* 90, 3 (1999), 333–347.

[15] David Graus, Paul N Bennett, Ryen W White, and Eric Horvitz. 2016. Analyzing and predicting task reminders. In *Proceedings of the Conference on User Modeling Adaptation and Personalization.* ACM, 7–15.

[16] Haibo He and Edwardo A Garcia. 2008. Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering* 9 (2008), 1263–1284.

[17] J Hill, LC Thomas, and DE Allen. 2000. Experts' estimates of task durations in software development projects. *International Journal of Project Management* 18, 1 (2000), 13–21.

[18] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems.* ACM, 159–166.

[19] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. ACM International Conference on Information and Knowledge Management (CIKM).

[20] Robert A Josephs and Eugene D Hahn. 1995. Bias and accuracy in estimates of task duration. *Organizational Behavior and Human Decision Processes* 61, 2 (1995), 202–213.

[21] Daniel Kahneman and Dan Lovallo. 1993. Timid choices and bold forecasts: A cognitive perspective on risk taking. *Management Science* 39, 1 (1993), 17–31.

[22] Daniel Kahneman and Amos Tversky. 1979. Intuitive prediction: biases and corrective procedures. *TIMS Studies in Management Science* 12 (1979), 313–327.

[23] Cornelius J König. 2005. Anchors distort estimates of expected duration. *Psychological Reports* 96, 2 (2005), 253–256.

[24] Cornelius J König, Andreja Wirz, Kevin E Thomas, and Rahel-Zoë Weidmann. 2015. The effects of previous misestimation of task duration on estimating future task duration. *Current Psychology* 34, 1 (2015), 1–13.

[25] Sander Koole and Mascha van't Spijker. 2000. Overcoming the planning fallacy through willpower: effects of implementation intentions on actual and predicted task-completion times. *European Journal of Social Psychology* 30, 6 (2000), 873–888.

[26] Justin Kruger and Matt Evans. 2004. If you don't want to be late, enumerate: Unpacking reduces the planning fallacy. *Journal of Experimental Social Psychology* 40, 5 (2004), 586–598.

[27] Nira Liberman and Yaacov Trope. 1998. The role of feasibility and desirability considerations in near and distant future decisions: A test of temporal construal theory. *Journal of Personality and Social Psychology* 75, 1 (1998), 5.

[28] Jingjing Liu, Chang Liu, Michael Cole, Nicholas J Belkin, and Xiangmin Zhang. 2012. Exploring and predicting search task difficulty. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management.* ACM, 1313–1322.

[29] 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.

[30] Don A Moore and Paul J Healy. 2008. The trouble with overconfidence. *Psychological Review* 115, 2 (2008), 502.

[31] Karen Myers, Pauline Berry, Jim Blythe, Ken Conley, Melinda Gervasio, Deborah L McGuinness, David Morley, Avi Pfeffer, Martha Pollack, and Milind Tambe. 2007. An intelligent personal assistant for task and time management. *AI Magazine* 28, 2 (2007), 47.

[32] Dat Quoc Nguyen, Dai Quoc Nguyen, Dang Duc Pham, and Son Bao Pham. 2014. RDRPOSTagger: A ripple down rules-based part-of-speech tagger. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics.* 17–20.

[33] Mark V Pezzo, Jordan A Litman, and Stephanie P Pezzo. 2006. On the distinction between yuppies and hippies: Individual differences in prediction biases for planning future tasks. *Personality and Individual Differences* 41, 7 (2006), 1359–1371.

[34] Ioannis Refanidis and Neil Yorke-Smith. 2010. A constraint-based approach to scheduling an individual's activities. *ACM Transactions on Intelligent Systems and Technology (TIST)* 1, 2 (2010), 12.

[35] Radim Rehurek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.*

[36] Xin Rong, Adam Fourney, Robin N Brewer, Meredith Ringel Morris, and Paul N Bennett. 2017. Managing uncertainty in time expressions for virtual assistants. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems.* ACM, 568–579.

[37] Michael M Roy and Nicholas JS Christenfeld. 2007. Bias in memory predicts bias in estimation of future task duration. *Memory and Cognition* 35, 3 (2007), 557–564.

[38] Michael M Roy and Nicholas JS Christenfeld. 2008. Effect of task length on remembered and predicted duration. *Psychonomic Bulletin and Review* 15, 1 (2008), 202–207.

[39] Michael M Roy, Scott T Mitten, and Nicholas JS Christenfeld. 2008. Correcting memory improves accuracy of predicted task duration. *Journal of Experimental Psychology: Applied* 14, 3 (2008), 266.

[40] Rafay A Siddiqui, Frank May, and Ashwani Monga. 2014. Reversals of task duration estimates: Thinking how rather than why shrinks duration estimates for simple tasks, but elongates estimates for complex tasks. *Journal of Experimental Social Psychology* 50 (2014), 184–189.

[41] Kevin Thomas, Simon Handley, and Stephen Newstead. 2004. The effects of prior experience on estimating the duration of simple tasks. *Current Psychology of Cognition* 22, 2 (2004), 83–100.

[42] Kevin E Thomas, Simon J Handley, and Stephen E Newstead. 2007. The role of prior task experience in temporal misestimation. *The Quarterly Journal of Experimental Psychology* 60, 2 (2007), 230–240.

[43] Yaacov Trope and Nira Liberman. 2000. Temporal construal and time-dependent changes in preference. *Journal of Personality and Social Psychology* 79, 6 (2000), 876.

[44] Dan Zakay and Richard A Block. 1996. The role of attention in time estimation processes. In *Advances in Psychology.* Vol. 115. Elsevier, 143–164.