Exploring Natural Language Processing Methods for Interactive Behaviour Modelling

Guanhua Zhang[1], Matteo Bortoletto[1], Zhiming Hu[1,2⋆], Lei Shi[1], Mihai Bâce[1], and Andreas Bulling[1]

[1]Institute for Visualisation and Interactive Systems, [2]Institute for Modelling and Simulation of Biomechanical Systems, University of Stuttgart, Stuttgart, Germany
{guanhua.zhang, matteo.bortoletto, zhiming.hu, lei.shi, mihai.bace, andreas.bulling}@vis.uni-stuttgart.de

Abstract. Analysing and modelling interactive behaviour is an important topic in human-computer interaction (HCI) and a key requirement for the development of intelligent interactive systems. Interactive behaviour has a sequential (actions happen one after another) and hierarchical (a sequence of actions forms an activity driven by interaction goals) structure, which may be similar to the structure of natural language. Designed based on such a structure, natural language processing (NLP) methods have achieved groundbreaking success in various downstream tasks. However, few works linked interactive behaviour with natural language. In this paper, we explore the similarity between interactive behaviour and natural language by applying an NLP method, byte pair encoding (BPE), to encode mouse and keyboard behaviour. We then analyse the vocabulary, i.e., the set of action sequences, learnt by BPE, as well as use the vocabulary to encode the input behaviour for interactive task recognition. An existing dataset collected in constrained lab settings and our novel out-of-the-lab dataset were used for evaluation. Results show that this natural language-inspired approach not only learns action sequences that reflect specific interaction goals, but also achieves higher F1 scores on task recognition than other methods. Our work reveals the similarity between interactive behaviour and natural language, and presents the potential of applying the new pack of methods that leverage insights from NLP to model interactive behaviour in HCI.

Keywords: Interactive Behaviour Modelling · Natural Language Processing · Mouse and Keyboard Input · Out-of-the-lab Dataset.

1 Introduction

Computational modelling of interactive behaviour has emerged as a key component of intelligent user interfaces (IUIs) in human-computer interaction (HCI) [66,70,3,2,14]. For example, understanding interactive behaviour helps HCI researchers and user experience (UX) designers analyse and improve interactive

[⋆] Corresponding author

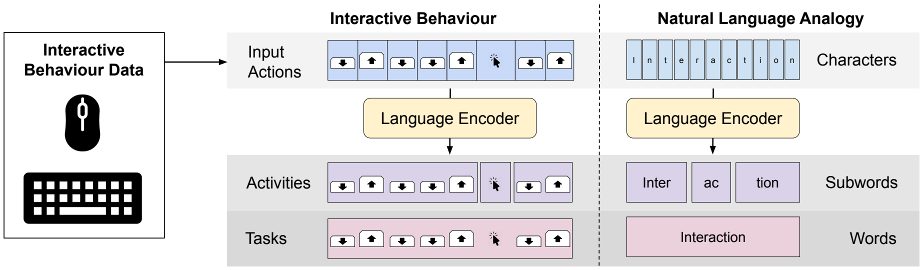


Fig. 1. Given that both interactive behaviour and natural language are sequential and hierarchical, we explored their similarity by applying an NLP method (a language encoder) to model mouse and keyboard behaviour.

systems [51,7]. Mouse and keyboard input is particularly promising because it is readily available on a large number of devices and pervasively used in daily life [66,59]. Interactive behaviour consists of low-level, atomic input actions that cannot be further decomposed [41], which may resemble characters in natural language. Furthermore, a sequence of such actions (an activity) that can reflect higher-level interaction goals may resemble a (sub)word that is a sequence of characters with semantic meanings. As such, interactive behaviour has both a sequential (actions happen one after another) and a hierarchical structure (a sequence of actions forms an activity driven by specific interaction goals), and hence may be similar to natural language (see Fig. 1). On the other hand, NLP methods, leveraging the sequential and hierarchical structure of input data, have recently achieved groundbreaking success in various downstream tasks like machine translation and question-answering [45,34,32,36]. However, analysing the possible similarity and link between interactive behaviour and natural language remains under-explored in HCI. One notable exception is the work by Han et al. that encoded n consecutive actions (like mouse clicks) into tokens to learn action embeddings [21]. However, at its core, the method uses n-gram, which limits the length of action sequences to a fixed length n and requires a dedicated search for its optimal value. Moreover, the vocabulary size grows exponentially as n increases [57]. Due to such drawback, n-gram has been dropped in NLP in favour of more flexible methods such as byte pair encoding (BPE) [48,47]. BPE and its variants are used in a significant number of large language models (LLMs) to encode text as subwords, allowing rare or unseen words to be handled without introducing new tokens every time [54,53]. Additionally, subwords in the vocabulary generated by BPE can have various lengths, allowing a rich and flexible vocabulary. In this work, we explore the similarity between mouse and keyboard behaviour and natural language, by using BPE to learn a vocabulary, i.e., a set of activities, which is further used to encode the behaviour to perform interactive task recognition. Knowing which task the user is conducting is essential for adaptive interactive systems that aim to understand interactive behaviour and interaction goals [44,17,26].

Existing mouse and keyboard datasets were typically collected in controlled laboratory settings, although behaviour tends to be more natural in out-of-the-lab settings [40]. We evaluate the method on two datasets that cover both settings and offer both modalities. For the lab setting, we chose the Buffalo dataset collected by Sun et al. [59] as it is the largest available dataset [43]. For the out-of-the-lab setting, given a lack of suitable publicly available data, we collected a novel multimodal dataset named EMAKI (Everyday Mouse And Keyboard Interactions)[1] EMAKI was collected from 39 participants performing three interactive tasks: *text entry and editing*, *image editing* and *questionnaire completion*. These tasks can be found in a wide range of applications and UIs, and cover varying types of mouse and keyboard actions.

On the two datasets, vocabulary analysis shows that BPE could learn explainable activities, e.g., reflecting graphical user interface (GUI) layouts and indicating interaction goals such as performing mouse dragging or keyboard shortcuts. Results from interactive task recognition show that BPE outperformed other methods on both modalities and datasets. In summary, our contributions are three-fold: (1) We collect EMAKI, a novel 39-participant out-of-the-lab mouse and keyboard dataset. (2) We explore the potential similarity between natural language and mouse and keyboard behaviour by learning meaningful activities via a commonly used NLP method, BPE. (3) We show that encoding with BPE also improves the performance of interactive task recognition. As such, our work uncovers the similarity between natural language and interactive behaviour, showing the potential for applying the new pack of methodology, i.e., NLP methods, to computational interactive behaviour modelling in HCI.

2 Related Work

2.1 Modelling Interactive Behaviour in HCI

Classical HCI approaches include descriptive models, e.g., Fitts's Law [1], and predictive models, e.g., the keystroke-level model (KLM) [12]. However, they are limited in strict controls and modelling simple tasks like pointing to a target or routine tasks that have to be specified step by step [12]. Recent research used 1D convolutional neural networks (CNN) [28,25], long short-term memory (LSTM) [26] and gated recurrent unit (GRU) [26] to encode gaze and head behaviour, based on the sequential structure, while others focused on spatial analysis and modelling [29,28]. Specifically, Xu et al. modelled mouse and keyboard behaviour by accumulating cursor positions into binary attention maps [66]. Other researchers modelled interactive behaviour from a statistical perspective. For example, Borji et al. used Hidden Markov Models (HMM) to encode motor actions including mouse clicks, mouse positions, and joystick positions in video games [8], while Sun et al. applied Gaussian mixture models (GMM) on keystrokes in text editing tasks [59]. Researchers also encoded eye movements [11]

[1] The dataset and code are available here: https://git.hcics.simtech.uni-stuttgart.de/public-projects/EMAKI

or gestures [63,55] into strings for activity recognition. Given that interactive behaviour has a sequential and hierarchical structure, which may resemble natural language, we explored modelling interactive behaviour from an NLP perspective.

2.2 Encoding Methods for Natural Language

Recent attractive success in NLP has been largely attributed to methods that efficiently encode characters [34], words [45] or sentences [50] into a vector representation. HCI researchers also followed this trend to model GUIs [38,61] or behavioural differences over time [21]. A key requirement for such methods is to encode or tokenise the input to generate a usable vocabulary of concepts. Due to the clear structure of natural language, NLP methods encode at the character, subword or word level. One popular approach is n-gram, which uses n words in a sequence to determine the context where commonly $n \leq 5$ [21,31,30,49]. However, such a method is limited by the choice of n, and the exponential increase of vocabulary size along n. More promising approaches learn a vocabulary of subwords, among which BPE has been widely used given that it allows rich and flexible vocabulary and understanding rare or unseen words [62,35,47,48]. Consequently, we employ BPE as the NLP method to create a vocabulary for interactive behaviour.

2.3 Analysis and Modelling of Mouse and Keyboard Behaviour

The mouse and keyboard are among the most widely used input modalities in daily interactions with computers [66,59]. Some researchers only focused on one modality, i.e., mouse or keyboard. Arapakis et al. explored different representations of mouse movements in web search tasks, including time series, heatmaps, and trajectory-based images [5], while Antal et al. employed 1D CNN to encode mouse actions including click and drag [3]. Dhakal et al. analysed keystroke patterns in a transcription typing task by correlation analysis [14], while Acien et al. employed LSTM to encode keystroke sequences in free text typing [2]. In contrast, Sun et al. explored both mouse and keyboard actions in two typing tasks, yet the work was limited to fully controlled laboratory settings [59].

3 Datasets for Evaluation

Although interactive behaviour, and specifically mouse and keyboard data, has been widely studied in HCI [59,66], most existing datasets have been collected in strictly controlled laboratory settings. Laboratory settings have the advantages of control and internal validity, but their ecological validity is highly limited [4]. Our out-of-the-lab data collection did not control where, when, how long and via which laptop or desktop participants could join, allowing more natural behaviour [40,46]. In addition, most datasets only include either mouse or keyboard data, while we opted for evaluations on both modalities. As such, we analysed mouse and keyboard behaviour from the in-the-lab Buffalo dataset [59] and

EMAKI, a novel multimodal out-of-the-lab dataset that we collected specifically for this purpose, given lacking suitable publicly available data. To evaluate constraints in data collection from a time perspective, *task* and *study* completion times were calculated. The former only counts the time spent on tasks, while the latter refers to finishing the entire study, including pauses.

3.1 The Buffalo Dataset

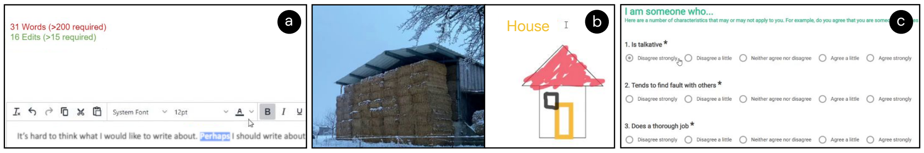
To the best of our knowledge, Buffalo [59] is the largest publicly available in-the-lab dataset containing both mouse and keyboard interactions. The dataset was collected with standalone keyboards over three sessions. 148 participants performed two typing tasks: transcribing a pre-defined text and typical office activities, such as answering predefined questions and sending emails. The average number of mouse actions and keystrokes per participant exceeded 19 K and 17 K, respectively. 75 participants completed both tasks with the same keyboard, while the remaining used three keyboards across sessions. Data from the former 75 participants were used in this work for a more controlled condition, following [65]. The average *task* completion time was 41.71 mins (SD = 6.34), while the average *study* completion time was slightly longer, 41.81 mins (SD = 6.27), indicating that participants barely took breaks in this constrained setting.

3.2 The EMAKI Dataset

We opted for an online study including three tasks: text entry and editing, image editing, and questionnaire completion. These tasks can be found in a wide range of interactive applications and UIs, and cover varying types of mouse and keyboard actions [66,59]. Furthermore, the tasks are neither limited to a particular real-world application [13,9] nor too controlled or artificial [14,70,71], different from the typing-focused tasks in Buffalo. Two short assessments were designed to analyse if participants show different proficiencies in using mouse and keyboard.

The study was implemented as a web application and hosted on our university server. The link to the study was sent directly to the participants. The frontend was implemented in JavaScript, while the backend consisted of a Node.js server and an SQLite database. We recorded clicks and key presses with separate events for press and release, mouse movements and their associated timestamps.

Participants We recruited 52 participants through university mailing lists and social networks. 12 participants who did not finish the study and one teenage participant were filtered out, leading to 39 participants in the end (18 female, 18 male and 3 “other gender”). Their ages ranged between 18 and 54 years ($\mathrm{M} = 25.05, \mathrm{SD} = 6.51$). Participants completed the study from 16 countries. On average, they reported having used mouse and keyboard for 13.64 years ($\mathrm{SD} = 6.80$). 15 participants used laptop touchpads, while the others used traditional mice. 28 participants used laptop keyboards and the rest used standalone keyboards.



31 Words (>200 required)
16 Edits (>15 required)

It's hard to think what I would like to write about. Perhaps I should write about

I am someone who...

Here are a number of characteristics that may or may not apply to you. For example, do you agree that you are some

1. Is talkative *

◉ Disagree strongly ○ Disagree a little ○ Neither agree nor disagree ○ Agree a little ○ Agree strongly

2. Tends to find fault with others *

○ Disagree strongly ○ Disagree a little ○ Neither agree nor disagree ○ Agree a little ○ Agree strongly

3. Does a thorough job *

○ Disagree strongly ○ Disagree a little ○ Neither agree nor disagree ○ Agree a little ○ Agree strongly

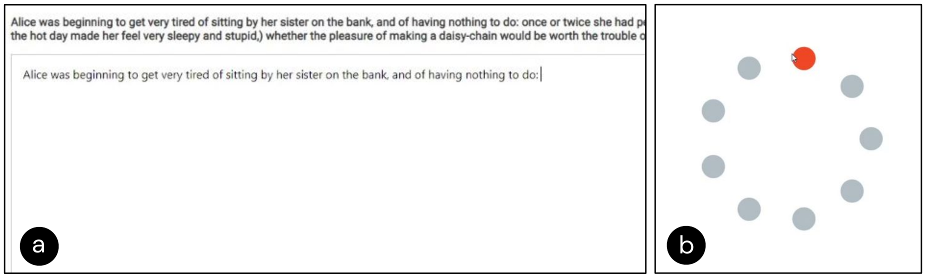
Fig. 2. Screenshots of the three interactive tasks in our online study: (a) text entry and editing, (b) image editing, and (c) questionnaire completion.

Interactive Tasks In task *text entry and editing*, participants wrote a piece of text in English in a text editor[2] for one trial (Fig. 2a). We did not specify the topic but offered suggestions, such as "summarise a movie/TV series/documentary that you recently watched" or "describe your pet". We asked participants to write ≥200 words and apply ≥15 formatting rules, e.g. change font size or alignment. We allowed any operation provided by the editor, such as copy-paste and undo. Two counters in the top left showed the number of words they already typed and formatting operations they applied. These counters were initially red and turned green once the minimum thresholds were reached.

In task *image editing*, participants were presented with two images shown side-by-side in an image editor[3] (Fig. 2b). The image on the left was a real photograph, whereas the image on the right was a sketch. On either or both sides, participants performed operations provided by the editor in any order they wanted. Candidate operations are drawing, cropping, flipping, rotating, adding icons and adding filters. To proceed to the next task, they had to perform at least 100 editing operations. In addition, we asked them to add at least one text box that contained a minimum of 10 characters. Similarly to the previous task, counters showed the task progress.

Questionnaire completion involved participants in completing four questionnaires[4], leading to four trials (Fig. 2c). These questionnaires served a dual purpose: providing information about participants, which can serve as metadata for future work on the dataset, while at the same time allowing us to record naturalistic mouse and keyboard data. The first questionnaire focused on demographics and included questions on gender, age, country of origin, country of residence, experience in using mouse and keyboard, and whether participants had any visual impairments. Afterwards were three widely-used personality questionnaires: BFI-44 (Big Five)[5], BIS-11 (Barratt Impulsiveness Scale)[6] and BIS-BAS (the Behavioural Inhibition and Approach System)[7].

[2] https://github.com/tinymce/tinymce
[3] https://github.com/nhn/tui.image-editor
[4] https://github.com/surveyjs/survey-library
[5] https://www.ocf.berkeley.edu/~johnlab/bfi.php
[6] http://www.impulsivity.org/measurement/bis11
[7] https://local.psy.miami.edu/people/faculty/ccarver/availbale-self-report-instruments/bisbas-scales/


Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had pe
the hot day made her feel very sleepy and stupid,) whether the pleasure of making a daisy-chain would be worth the trouble o

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do:

Fig. 3. Two proficiency assessments: (a) text typing and (b) move and click.

Procedure Before starting with the tasks, participants were asked to carefully read the study goals and task descriptions. They were then asked whether they were using a mouse or touchpad, and a laptop or standalone keyboard. To start the study, participants had to click two checkboxes to confirm that (1) they had read and understood the goals of the study, and (2) their data may be published and analysed for research purposes. Afterwards, participants performed tasks in fullscreen. If they left the fullscreen mode during a task, the task was restarted. We opted for the design to discourage participants from multitasking. To reduce potential effects of task order, half of the initial 52 participants performed the text entry and editing task first, followed by the image editing task, while the other half performed in the inverse order. After data filtering, 24 participants did the text task and then image task, while the other 15 in the inverse order. We always showed questionnaires at the end, following studies that also collected personality questionnaires [24,42]. Detailed guidelines for tasks were available to participants throughout the study. Participants could contact us whenever they had questions, felt uncomfortable or unsure of any task or wanted to withdraw. Upon completion of the study, participants were shown their results of personality questionnaires as compensation. No monetary compensation was made.

Dataset Statistics The average task completion time was 37.40 mins (SD = 13.91), in which 16.60 mins (SD = 8.51) were spent on text entry and editing, 6.15 mins (SD = 3.60) on image editing, and 9.84 mins (SD = 4.48) on questionnaires. The average study completion time was significantly longer, 55.33 mins (SD = 29.32). In total, we collected 1.14 M mouse actions and 205 K keyboard actions. 38% of mouse actions were generated from the image editing task, 43% from questionnaire completion, while only 19% came from the text entry and editing task. Text entry and editing contributed 92% of the keyboard actions, while only 8% were from the other two tasks (image editing: 3%, questionnaire completion: 5%).

Assessments of Proficiency Before interactive tasks, our study also included two short assessments to analyse if participants who used different types of input devices showed different proficiencies in using mouse and keyboard. The two

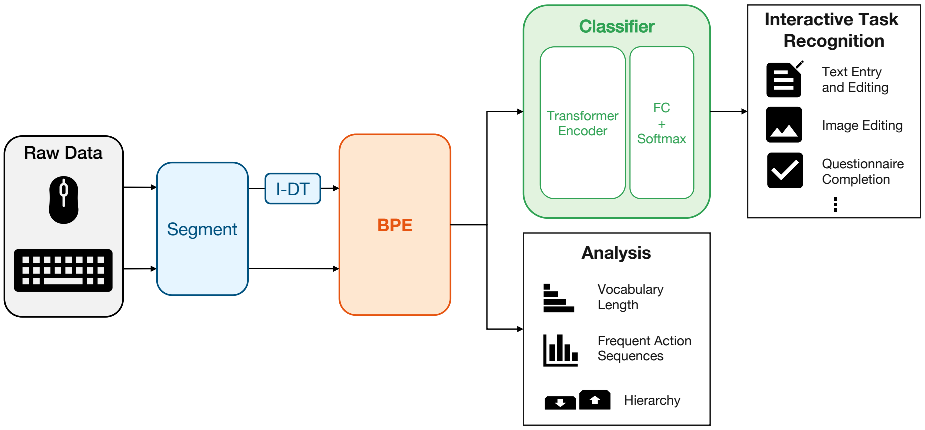


Fig. 4. Overview of our pipeline of exploring modelling interactive behaviour from an NLP perspective.

assessments were *text typing* for keyboard proficiency and *move and click* for mouse proficiency, shown in Fig. 3. *Text typing* involved copying a short piece of text (∼100 words, Fig. 3a) as quickly as possible [19]. The average duration of key presses and the number of keys pressed per minute were calculated as keyboard metrics [19]. *Move and click* was inspired by a Fitts's Law task [56], where participants clicked an orange dot that randomly appeared at a predefined location as quickly as possible over multiple rounds. Once clicked, the orange dot turned grey and another random dot turned orange (Fig. 3b). Fitts's law [15] models movement time as $MT = a + b\log_2\left(\frac{2d}{w}\right)$, where d is the distance between the centre of the target and the starting point; w is the width of the target; a and b are constants that can be interpreted as the delay and the acceleration. Based on d, w and MT recorded in *move and click*, we computed a and b via linear regression and used them as metrics of mouse proficiency.

Based on the type of mouse (touchpad vs. traditional mouse), we split participants into two groups and then calculated mouse metrics from data collected in the mouse assessment. A Mann-Whitney U test showed that both metrics were significantly different between the two groups. One reason is that touchpad and traditional mouse lead to different pointing speeds and accuracies [23]. Then, we split participants into two groups based on using a laptop or standalone keyboard. No significant difference was found in keyboard metrics calculated from the keyboard assessment.

4 Modelling Interactive Behaviour with an NLP Method

As Fig. 4 shows, the raw data (mouse and keyboard action sequences) are first segmented into subsequences. Core to our approach is BPE learning a vocabulary of subwords, i.e. a set of meaningful mouse and keyboard activities, and then encoding the behaviour based on the vocabulary. As BPE requires discrete inputs, mouse data are preprocessed additionally using the dispersion-threshold identification (I-DT) algorithm, that converts continuous-valued mouse coordinates into discrete tokens. The encodings generated by BPE are then evaluated in two ways to explore if a natural language-like structure exists in mouse and keyboard behaviour that can be captured by this widely used NLP method: (1) analyse the semantic meaning of the vocabulary, i.e., interaction goals underlying learnt activities, and (2) as input to train a Transformer-based classifier for task recognition. The two evaluations are demonstrated in Section 5.

4.1 Data Preprocessing

Different from natural language where words and sentences are separated by spaces and punctuations, modelling interactive behaviour first requires splitting data into smaller units. Thus, a sliding non-overlapping window was used to segment the long raw data. On the keyboard actions, the window lengths L_{win} were empirically set to 10, 50, and 100. The window lengths L_{win} for the mouse actions were set to 20, 100 and 200, as we observed on both datasets that, the number of generated mouse actions for a fixed time window is roughly twice as many as the keyboard actions. When using both modalities jointly, the window lengths were set to the mean value of those for single modalities, i.e. L_{win} = 15, 75 and 150. For keyboard actions, the action type and the key value were concatenated as a token, e.g., *KeyDown_a* (a↓) or *KeyUp_Shift* (Shift↑). Buffalo recorded 91 key values, while EMAKI had 137 values, yielding 182 and 274 atomic actions forming the starting vocabulary, respectively. With more types of keys, EMAKI can potentially reflect more behaviour varieties.

Participants completed our study on their own computers with different screen resolutions, so we first re-scaled the mouse coordinates to $[0, 1]$. For consistency, we re-scaled Buffalo mouse data to the same range. We observed two categories of mouse behaviour: *pinpoint*, i.e. interacting with the target UI element in a small area, where moves are shorter, slower and more concentrated, resembling gaze fixations; and *re-direction* between targets, resembling fast saccadic eye movements between fixations [52]. Inspired by gaze fixation detection, we used I-DT [52] to preprocess mouse data (see Appendix). Then we divided the screen equally into four areas (0: top-left, 1: top-right, 2: bottom-left, 3: bottom-right). The action type (move or click), mouse behaviour category (pinpoint or re-direction), and the screen area were concatenated as a token, e.g., *Move_Redirection_Area0* or *Click_Pinpoint_Area3*. When representing clicks, Buffalo only recorded a *Click*, while we recorded both *Down* (press) and *Up* (release) events. Therefore, Buffalo has 2×2×4=16 atomic actions and EMAKI has 3×2×4=24.

4.2 Encoding Mouse and Keyboard Behaviour with BPE

We employed BPE (see Appendix for its algorithm) to learn a vocabulary of subwords, i.e., activities that consist of various numbers of consecutive actions.

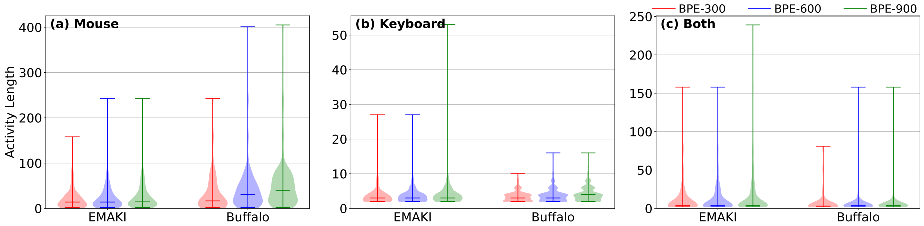


Fig. 5. Violin plots for the lengths of activities learnt by BPE after 300 (in red), 600 (in blue) and 900 (in green) iterations, of (a) mouse, (b) keyboard and (c) both modalities on EMAKI and Buffalo datasets. Each bar shows the range of activity lengths, while the middle line indicates the median length. The y-axes are scaled according to the range in each subplot.

Starting from the action sequence set D, the vocabulary V is built after k iterations. In each iteration, the most frequent pair of actions or activities form a new activity, which is added into V and used to update D. We consider each action as a character, given it is an inseparable, atomic unit. The initial vocabulary is composed of actions and one extra token representing the end of the action sequence from one task trial. Thus, the initial vocabulary sizes are $|V|_{\text{mouse}} = 17$ and $|V|_{\text{key}} = 183$ in Buffalo, and $|V|_{\text{mouse}} = 25$, $|V|_{\text{key}} = 275$ in EMAKI. We set k to 300, 600 and 900 empirically.

5 Evaluations of the NLP Method

As mentioned at the beginning of Section 4, BPE was evaluated in two ways: (1) we analysed its vocabulary to examine if the way of learning semantic subwords from characters could learn meaningful activities from interactive actions; and (2) we tested if encoding interactive behaviour in this NLP fashion benefited a downstream task, interactive task recognition, using a Transformer-based classifier.

5.1 Analysis of the Learnt Vocabulary

We first examined statistics of the vocabulary including its size and activity lengths. Then we analysed semantic meanings of the most frequent and long activities. Frequent activities are short, low-level and pervasively exist in various activities, while long activities reflect high-level and complex goals.

Vocabulary Statistics. As Fig. 5a shows, in EMAKI the maximum length of mouse activities reached 243 actions (BPE-900), while the median length was 16. The longest keyboard activity had 53 actions, while the median length was 3 (Fig. 5b). When using both modalities jointly, the maximum activity length was 239 after 900 iterations, while the median length was 4 (Fig. 5c). In Buffalo, the lengths of activities had a maximum of 405 and a median of 39 from mouse behaviour (Fig. 5a); a maximum of 16 and a median of 4 from keyboard behaviour (Fig. 5b); and a maximum of 158 and a median of 4 from

Dataset	EMAKI			Buffalo		
Method	BPE-300	BPE-600	BPE-900	BPE-300	BPE-600	BPE-900
Mouse	322	622	921	310	609	909
Keyboard	513	808	1103	473	770	1067
Both	569	864	1163	496	790	1084

Table 1. Vocabulary sizes generated using BPE after 300, 600 and 900 iterations, on EMAKI and Buffalo datasets.

Rank	1	2	3	4	5	6	7	8	9	10
EMAKI	⊔↓, ⊔↑	⇐↓,⇐↑	e↓, e↑	t↓, t↑	a↓, a↑	o↓, o↑	⇐↓,⇐↑,⇐↓,⇐↑	i↓, i↑	s↓, s↑	n↓, n↑
Buffalo	⊔↓, ⊔↑	⇐↓,⇐↑	e↓, e↑	t↓, t↑	o↓, o↑	i↓, i↑	a↓, a↑	s↓, s↑	n↓, n↑	l↓, l↑

Table 2. The ten most frequent keyboard activities found by BPE. The ⊔ symbol represents Space. The ⇐ symbol means Backspace. The down arrow ↓ and up arrow ↑ denote *KeyDown* and *KeyUp*, respectively.

joint modalities (Fig. 5c). Mouse activities were longer than keyboard activities, indicating that the preprocessed mouse data were more similar compared to preprocessed keyboard data. Comparisons between datasets show that mouse activities in EMAKI were more diverse, while Buffalo contained more diverse keyboard activities.

Table 1 shows the vocabulary sizes generated by BPE on the two datasets. Note that starting from BPE-k and running the algorithm for k more iterations, the vocabulary size increases by approximately k elements – showing that BPE overcomes the issue of exponential growth of vocabulary size in n-gram.

Frequent Activities. The three BPE iterations learnt the same top-10 frequent keyboard action sequences as shown in Table 2. Eight out of ten action sequences are the same on the two datasets, although they were collected from different participants in different experimental settings, indicating that generalised patterns underlie keyboard behaviour. The interaction goal behind the most frequent activity is to press the spacebar, which is in line with the observation that spaces occur often when typing in various languages. The second frequent activity reflects an intention of pressing Backspace which is frequently and widely used to correct what has been typed. Most frequent activities correspond to character keystrokes, and reflect the top-7 most frequent English letters: “e” (12.15%),“a” (8.67%), “t” (8.60%), “i” (7.53%), “o” (7.38%), “n” (7.34%) and “s” (6.63%) [20]. The difference in their order may be due to that the datasets are limited to specific typing scenarios and not representative of the entire English language. We also noticed that the left and right arrows, for redirecting typing locations, were also frequent on both datasets.

The most frequent ten mouse action sequences learnt by BPE were also the same on the two datasets. All of them are mouse moves of pinpoint, implying that participants follow similar ways to interact with UI targets even in different tasks and settings. These pinpointing regions were primarily in the top-left and bottom-left areas, while fewer pinpoints fell on the right side. This matches the layouts of not only general GUIs but also those used in our user study. For example, menu bars and sidebars are commonly at the top and to the left of interactive windows, respectively. Also, our text formatting tools were at the top of the text editor. The image editing tools were in the leftmost of the image editor. Additionally, our questionnaires were left-aligned, so the choices for participants to click lay to the left.

Interaction Goals behind Activities. We also analysed long activities to examine if BPE learnt a hierarchy, i.e., if atomic actions form meaningful activities driven by complex goals. An example is "Dot↓, Dot↑, Space↓, Space↑, Shift↓, i↓, i↑, Shift↑, Space↓, Space↑". The goal behind the whole sequence is to start a sentence with the word "I", in line with the common texting or typing scenario of writing about oneself. It consisted of the low-level goal of pressing each aforementioned key, which was further composed of atomic actions *KeyDown* and *KeyUp*. BPE also learnt "Space↓, Space↑, Backspace↓, Backspace↑" from EMAKI, suggesting that participants typed at a faster pace than their thought process. Another example is "Ctrl↓, s↓, s↑, Ctrl↑" from Buffalo, representing the shortcut for saving files. Looking at mouse behaviour, BPE captured drag behaviour, represented as a *MouseDown* action followed by multiple *MouseMove* actions and ending with a *MouseUp* action. Another learnt long activity had 37 actions with 35 moves and a click as pinpoint in area 0, reflecting the goal of adjusting the cursor to a target and then clicking.

5.2 Interactive Task Recognition

We also evaluated the practical effectiveness of our approach on interactive task recognition. Knowing which task a user is performing enables adaptive UIs to understand the interactive behaviour and goals [26,27]. We compared our approach with two baselines: an ablated version which bypasses encoding (noted as NoEncoding) and replacing BPE with an autoencoder (AE). Autoencoder, consisting of an encoder and a decoder, is trained in a self-supervised way to reconstruct the input with the lowest error. Therefore, it needs no annotations and has a high generalisability, also used on language data [37]. To control variables, i.e., restrict the comparison to the encoding, we set two rules: (1) to reduce the impact of sophisticated designs of the encoders, use vanilla AE and BPE; (2) use the same hyperparameter sets for the classifier.

We implemented an AE that includes four components: an embedding layer of dimension $d_e = 128$ to handle discrete tokens; an encoder component composed of one to three fully connected (FC) layers with hidden dimensions (64), $(64, 32)$ and $(64, 32, 16)$; a decoder component, which is symmetric to the encoder; and a reconstruction component consisting of an FC layer and a softmax layer. Dropout was added after FC layers to avoid overfitting. We denote the autoencoder that has one, two, or three FC layers in the encoder and decoder components as AE-1, AE-2 and AE-3. Cross entropy between the reconstructed sequences and the input was used as the loss function. After training, the encoder component was used to encode interactive behaviour.

Our task classifier is based on a Transformer [60], which is well known for its success in NLP and capability to handle long dependencies in temporal signals. The classifier is composed of $N = \{2, 4, 6\}$ Transformer encoder layers, then an FC and softmax layer. Each Transformer encoder layer had $h = 4$ attention heads, $d_{\text{model}} = \{16, 64\}$ expected features, $d_{\text{ff}} = 4d_{\text{model}}$ dimension in feedforward layers and uses the ReLU activation function. During training, we applied label smoothing with $\epsilon = 0.1$ [60]. We used AdamW optimizer with learning rate $lr = \{10^{-3}, 10^{-4}\}$ and $\beta = (0.9, 0.999)$ [10] and the cross entropy as loss function. The training was done on a Tesla V100 GPU with a batch size of 64 and a dropout rate of 0.5. The classifier was trained for 30 epochs, while the AE was trained for 10 epochs because of its faster convergence. Because activities in the flexible vocabulary learnt by BPE have different lengths, we padded short samples and applied padding masks.

EMAKI has three main interactive tasks, posing a three-class classification problem, while Buffalo has two tasks, posing a binary classification problem. The evaluation follows 5-fold participant-independent cross-validation, where data from 80% of participants form the training set and the remaining participants form the test set. This scheme can evaluate the performance of unseen users. Macro F1 score [24] was chosen as evaluation metric because of the imbalanced classes, e.g., most keyboard data were from the text task on EMAKI. For each model, we report the highest F1 score achieved among all the parameter sets. Results show that on both datasets methods using BPE encoding outperformed the others (see Fig. 6 and 7).

Results on EMAKI On mouse data, BPE-300 consistently outperformed other methods (Fig. 6a). A one-way ANOVA test showed that differences between methods are significant (p<.001): F=9.697 on L_{win}=200, F=12.396 on L_{win}=100 and F=7.194 on L_{win}=20. A post-hoc Tukey HSD test further confirmed that BPE-300 significantly outperformed the other methods on L_{win}=200, L_{win}=100 (p<.001 for AE and p<.05 for NoEncoding) and L_{win}=20 (p<.01 for both AE and NoEncoding). Fig. 6b shows that BPE-600 achieved the best results for L_{win}=100 and L_{win}=50, whereas when L_{win}=10 the best was BPE-300. Differences between methods are significant (F=13.044, p<.001 for L_{win}=100, F=4.620, p<.01 for L_{win}=50 and F=4.220, p<.01 for L_{win}=10). Post-hoc Tukey HSD tests confirmed that BPE-600 significantly outperformed NoEncoding (p<.01) and AE-1 (p<.001) for L_{win}=100. On joint modalities, BPE-300 performed the best, with the highest F1 score of 0.693 (Fig. 6c). Differences between methods were again significant with F=13.996, p<.001 on L_{win}=150, F=5.678, p<.001 on L_{win}=75 and F=2.665, p<.05 on L_{win}=15. Tukey HSD test indicated that BPE-300 significantly outperformed AE (p<.01) and NoEncoding (p<.05) on L_{win}=150 and both of them at p<.05 when L_{win}=75.

In Section 3.2, we report that participants using touchpads and traditional mice show different proficiencies. Therefore, we analysed if such differences affected task recognition. We separately performed 5-fold cross-validation based on the two groups. Since 24 participants used traditional mice while only 15 used touchpads, we randomly selected 15 traditional mouse users to reduce the

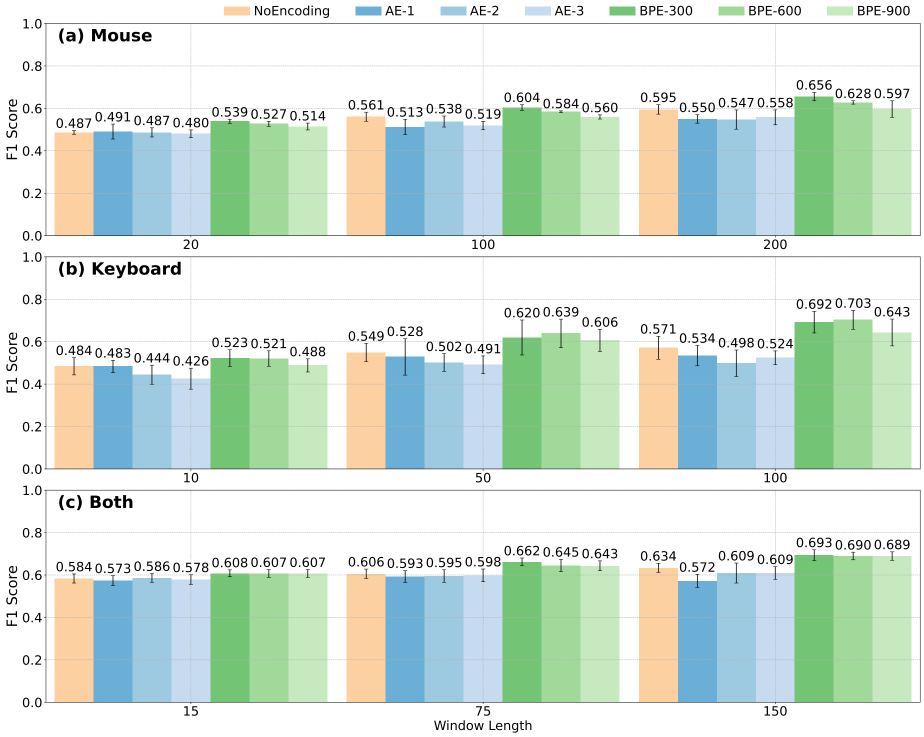


Fig. 6. F1 scores of recognising three interactive tasks on EMAKI from (a) mouse, (b) keyboard and (c) both modalities, segmented by different windows. Error bars represent the standard deviation from a 5-fold cross-validation.

influence of data amount on performance. Because BPE-300 on the longest window achieved the best results on mouse data (Fig. 6a), we used the same setting and did a Mann-Whitney U test on F1 scores achieved from two groups. To mitigate the randomisation introduced by participant selection, we repeated the above procedure five times. None of the five tests found a significant difference in performance. The reason may be that our method does not explicitly encode time information, thus ignoring the speed difference in moving the cursor [23].

Results on Buffalo On mouse data (Fig. 7a), BPE-300 performed the best and got the highest F1 score of 0.547. One-way ANOVA showed that differences between methods were significant (p<.001) with F=20.345 for L_{win}=200, F=18.609 for L_{win}=100 and F=5.589 for L_{win}=20). Post-hoc Tukey HSD tests showed that BPE-300 significantly outperformed NoEncoding (p<.05) and AE-1 (p<.001) when L_{win}=200. On keyboard data (Fig. 7b), BPE-900 and BPE-600 outperformed other methods. Differences between methods are significant with F=30.218 for L_{win}=100, F=5.884 for L_{win}=50 (both p<.001) and F=4.791, p<.01 for L_{win}=10. According to post-hoc Tukey HSD tests, BPE-900 signifi-

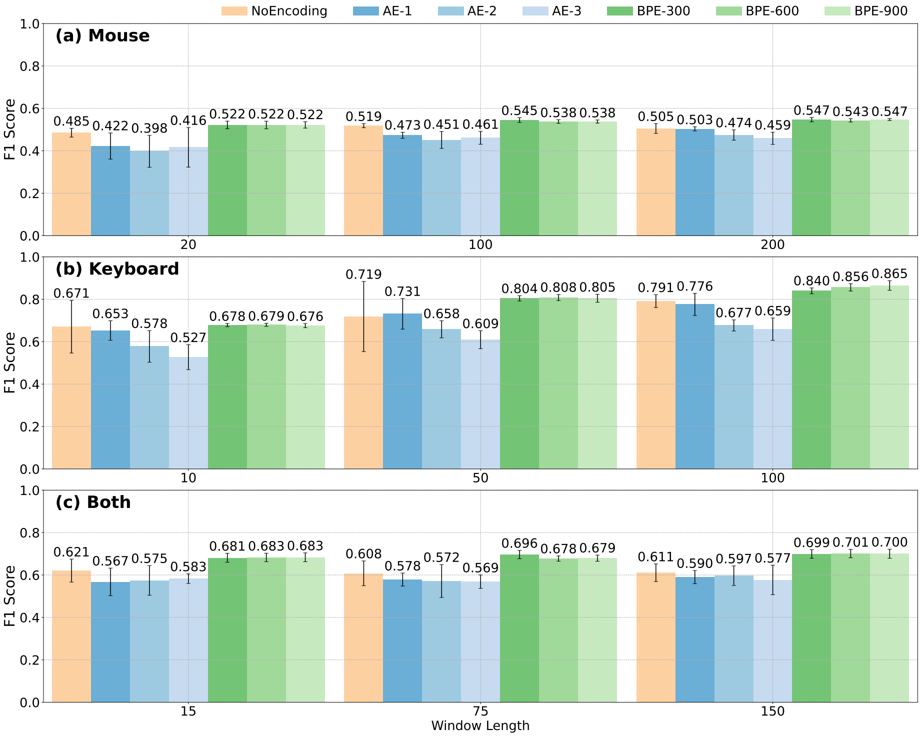


Fig. 7. F1 scores of recognising two interactive tasks on Buffalo from (a) mouse, (b) keyboard and (c) both modalities, segmented by different windows. Error bars represent the standard deviation from a 5-fold cross-validation.

cantly outperformed AE-1 (p<.01) and NoEncoding (p<.05) when L_{win}=100, and BPE-600 significantly outperformed AE-1 (p<.001) when L_{win}=50. On joint modalities (Fig. 7c), BPE resulted in similar yet higher F1 scores than baselines. The best result was achieved by BPE-600 on the longest window of 0.701. Differences between methods were again significant (p<.001): F=10.733 for L_{win}=150; F=11.151 for L_{win}=75; and F=7.397 for L_{win}=15. Tukey HSD test showed that BPE-600 significantly outperformed AE-2 (p<.01) and NoEncoding (p<.05) on L_{win}=150; BPE-300 outperformed AE-1 (p<.01) and NoEncoding (p<.05) on L_{win}=75; and BPE-600 outperformed AE-3 (p<.05) on L_{win}=15.

It is noticeable that results obtained from Buffalo mouse data slightly exceeded the chance level and were much worse than those from keyboard data. A possible reason is that the mouse behaviour on the Buffalo dataset was similar across different tasks. To verify this, we calculated the average distances between mouse trajectories in different interactive tasks, following [64]: (1) all the mouse actions generated in one trial by one participant were considered one trajectory, on which 101 points were sampled uniformly; (2) the distance between two trials was defined as the average Euclidean distance between each pair of points on

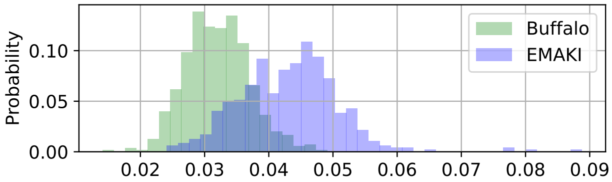


Fig. 8. Distribution of the average Euclidean distances between mouse trajectories from different interactive tasks on the two datasets. Smaller distances mean that trajectories from different tasks are more similar.

two trajectories; (3) the distance between two tasks was computed as the average distance between each trial from task 1 and each from task 2. Fig. 8 shows that the distance between tasks from Buffalo is smaller than EMAKI, suggesting that mouse behaviour generated from the two tasks from Buffalo is similar, consistent with the statistics of BPE vocabulary (Section 5.1). Therefore, it is more difficult to classify tasks based on Buffalo mouse data.

6 Discussion

6.1 Modelling Interactive Behaviour from a Natural Language Perspective

Our work is among the first to explore the similarity between interactive behaviour and natural language, given that both have a sequential and hierarchical structure. Towards this goal, we applied BPE, which has been commonly used in state-of-the-art large language models to encode mouse and keyboard behaviour. At the lowest level, input actions were considered as characters since they are atomic and inseparable. For higher levels, BPE learned “subwords” from interactive behaviour, which were interactive activities, i.e., action sequences driven by underlying interaction goals. The analysis of the learnt vocabulary showed that following the same way of learning the semantic hierarchy of language, BPE was able to capture meaningful activities such as mouse drags, keyboard shortcuts and precisely adjusting the mouse to click on a UI element (Section 5.1). Despite representing just a first exploration, the insights from our analysis underline the similarity between interactive behaviour and natural language, and indicate the possibility of applying more powerful NLP methods like BERT [32,39] to encode interactive behaviour. Besides the state-of-the-art performances achieved, such LLMs also have noticeable advantages of generalisability and reusability. They can be pretrained on one dataset and re-used to encode other datasets to solve various downstream tasks with fine-tuning, which is more cost-effective than dedicating a specific large model towards each dataset or task [58]. Future HCI research can follow such NLP methods to build reusable pretrained interactive behaviour models for better generalisability and cost-effectiveness.

6.2 NLP Encoding for Interactive Task Recognition

Interactive task recognition is one of the key requirements of intelligent interactive systems to understand and adapt to interactive behaviour and interaction goals [44,17,18,33]. On this recognition task, encoding with BPE significantly outperformed baselines on both datasets, all the modalities and windows. Specifically, on our out-of-the-lab, newly collected EMAKI dataset, encoding with BPE obtained the highest F1 score of 0.703 recognising three tasks (Fig. 6). On average, BPE improved the F1 score by 0.087 on keyboard data, 0.051 on mouse, and 0.044 on the joint modalities. On the Buffalo dataset, BPE achieved the highest F1 score of 0.865 (Fig. 7) recognising two tasks. On average, BPE improved the F1 score by 0.080 on joint modalities, 0.053 on keyboard data and 0.035 on mouse data. These results, from a practical perspective, further reveal the promising effectiveness of modelling interactive behaviour as natural language.

We observed that methods generally achieved better results on longer windows, which may be due to that more actions may uncover richer characteristics of the tasks. However, increasing the window size yields fewer training samples and makes the recognition model wait longer for a complete window of actions to provide a prediction. In our experiments, windows that led to the best performance on mouse, keyboard and joint modalities had 200, 100 and 150 actions, respectively. These values can be a reference for future mouse and keyboard behaviour modelling methods.

In addition, on both datasets, using BPE on keyboard behaviour improved the F1 score more than on mouse behaviour, indicating its better ability of handling keyboard than mouse behaviour. This finding is expected, as typing on a keyboard is directly linked to expressing natural language. A second reason might be that discretising mouse data caused a loss of information [67,68]. On the joint modalities, we observed a general performance improvement from individual modalities on EMAKI, but not on Buffalo. As shown in Section 5.2, Buffalo lacks the diversity in mouse behaviour and thus performance achieved by combining mouse and keyboard is in-between that of individual modality.

6.3 EMAKI Dataset

Most publicly available mouse and keyboard datasets were collected in constrained laboratory settings, such as the Buffalo dataset. In contrast, our EMAKI is a step towards fully unconstrained settings to allow more natural interactive behaviour. Our study did not control where, when, or how long participants joined the study. In addition, participants used their own devices, which contributes to ecological validity. Consequently, our participant pool is more diverse given that participants are from different countries, and used different input devices and screen resolutions. All of Buffalo's participants were university students between 20-30 years old, while ours were between 18-54 and covered non-student participants. Moreover, our participants spent various time on tasks as they were freer to pause and resume (as shown in Section 3.1 and 3.2). Buffalo primarily uses typing-focused tasks, while EMAKI has complementary characteristics and tasks – like image editing and questionnaires – encouraging diverse mouse behaviour, as confirmed by our analysis in Section 5.1. Furthermore, higher diversity in behaviour can lead to better task recognition performance (Section 5.2). We also verified that the amount of data in EMAKI is sufficient for training the method for task recognition (see Appendix). Besides serving as a benchmark for task recognition, the questionnaires included in EMAKI also encourage future research on the interplay between multimodal behaviour and personality traits [71].

6.4 Limitations and Future Work

Our user study covered diverse but predefined tasks and did not allow multitasking. In the future, we will move towards fully uncontrolled settings. Time information may further improve behaviour modelling [6,16] and will be explicitly encoded in future work. We chose BPE over N-gram due to its flexibility, yet for systems where activities have similar lengths, N-gram might be efficient enough. An interesting future work is to explore the boundary of where the methods lead over the other. Also, even used on both modalities jointly, BPE learned activities composed of single modalities. A possible reason is that the behaviour of switching between mouse and keyboard is diverse, which BPE could not capture. Future work can explore the use of other NLP methods to better learn the interplay between mouse and keyboard behaviour [39,48,47]. Automatic interpreters can be studied to identify meaningful and interesting insights into behaviour from the BPE vocabulary, instead of human interpretation. Moreover, we intend to study other interactive modalities, such as screen touch and mid-air gesture, as well as other HCI downstream tasks like personality recognition.

7 Conclusion

We explored the similarity between interactive behaviour and natural language, given that both of them have a sequential and hierarchical structure. Towards the goal, we applied a widely used NLP method, BPE, to encode mouse and keyboard behaviour by learning its subwords, i.e., activities. Results on an existing controlled dataset and a novel out-of-the-lab dataset showed that the method can capture meaningful activities. Moreover, encoding with BPE significantly improved interactive task recognition, which is commonly required in intelligent interactive systems. Taken together, our exploratory work links interactive behaviour with natural language and provides a promising NLP perspective for modelling interactive behaviour, which has the potential to improve the generalisability of computational interactive behaviour models (Section 6.1) and also performances of interactive behaviour-based HCI tasks.

Acknowledgement

The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting G. Zhang. M. Bortoletto and A. Bulling were funded by the European Research Council (ERC; grant agreement 801708). Z. Hu and L. Shi were funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2075 – 390740016. M. Bâce was funded by a Swiss National Science Foundation (SNSF) Postdoc.Mobility Fellowship (grant number 214434). We acknowledge the support by the Stuttgart Center for Simulation Science (SimTech). We would like to thank Yuanfan Ying and Dominike Thomas for their helpful suggestions. Lastly, we would like to thank the anonymous reviewers for their useful feedback.

References

1. Accot, J., Zhai, S.: Beyond fitts' law: models for trajectory-based hci tasks. In: Proceedings of the ACM SIGCHI Conference on Human factors in computing systems. pp. 295–302 (1997)
2. Acien, A., Morales, A., Vera-Rodriguez, R., Fierrez, J., Monaco, J.V.: Typenet: Scaling up keystroke biometrics. In: Proceedings of the 2020 IEEE International Joint Conference on Biometrics. pp. 1–7. IEEE (2020)
3. Antal, M., Fejér, N., Buza, K.: Sapimouse: Mouse dynamics-based user authentication using deep feature learning. In: Proceedings of the 2021 IEEE International Symposium on Applied Computational Intelligence and Informatics. pp. 61–66. IEEE (2021)
4. Apaolaza, A., Harper, S., Jay, C.: Understanding users in the wild. In: Proceedings of the 10th international cross-disciplinary conference on web accessibility. pp. 1–4 (2013)
5. Arapakis, I., Leiva, L.A.: Learning Efficient Representations of Mouse Movements to Predict User Attention, p. 1309–1318. Association for Computing Machinery, New York, NY, USA (2020)
6. Azcarraga, J.J., Ibañez, J.F., Lim, I.R., Lumanas Jr, N.: Use of personality profile in predicting academic emotion based on brainwaves signals and mouse behavior. In: 2011 Third International Conference on Knowledge and Systems Engineering. pp. 239–244. IEEE (2011)
7. Bi, X., Smith, B.A., Zhai, S.: Multilingual touchscreen keyboard design and optimization. Human–Computer Interaction **27**(4), 352–382 (2012)
8. Borji, A., Sihite, D.N., Itti, L.: Probabilistic learning of task-specific visual attention. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 470–477. IEEE (2012)
9. Brown, E.T., Ottley, A., Zhao, H., Lin, Q., Souvenir, R., Endert, A., Chang, R.: Finding waldo: Learning about users from their interactions. IEEE Transactions on visualization and computer graphics **20**(12), 1663–1672 (2014)
10. Brückner, L., Leiva, L.A., Oulasvirta, A.: Learning gui completions with user-defined constraints. ACM Transactions on Interactive Intelligent Systems (TiiS) **12**(1), 1–40 (2022)
11. Bulling, A., Ward, J.A., Gellersen, H., Tröster, G.: Robust Recognition of Reading Activity in Transit Using Wearable Electrooculography. In: Proc. International Conference on Pervasive Computing (Pervasive). pp. 19–37 (2008). https://doi.org/10.1007/978-3-540-79576-6_2
12. Card, S.K., Moran, T.P., Newell, A.: The keystroke-level model for user performance time with interactive systems. Communications of the ACM **23**(7), 396–410 (1980)

13. Chudá, D., Krátky, P.: Usage of computer mouse characteristics for identification in web browsing. In: Proceedings of the 15th International Conference on Computer Systems and Technologies. pp. 218–225 (2014)
14. Dhakal, V., Feit, A.M., Kristensson, P.O., Oulasvirta, A.: Observations on typing from 136 million keystrokes. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. p. 1–12. CHI '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3173574.3174220
15. Fitts, P.M.: The information capacity of the human motor system in controlling the amplitude of movement. Journal of experimental psychology **47**(6), 381 (1954)
16. Freihaut, P., Göritz, A.S.: Does peoples' keyboard typing reflect their stress level? an exploratory study. Zeitschrift für Psychologie **229**(4), 245 (2021)
17. Fu, E.Y., Kwok, T.C., Wu, E.Y., Leong, H.V., Ngai, G., Chan, S.C.: Your mouse reveals your next activity: towards predicting user intention from mouse interaction. In: 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC). vol. 1, pp. 869–874. IEEE (2017)
18. Gajos, K., Weld, D.S.: Supple: automatically generating user interfaces. In: Proceedings of the 9th international conference on Intelligent user interfaces. pp. 93–100 (2004)
19. Grabowski, J.: The internal structure of university student's keyboard skills. Journal of writing research **1**(1), 27–52 (2008)
20. Grigas, G., Juškevičienė, A.: Letter frequency analysis of languages using latin alphabet. International Linguistics Research **1**(1), p18–p18 (2018)
21. Han, L., Checco, A., Difallah, D., Demartini, G., Sadiq, S.: Modelling user behavior dynamics with embeddings. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. pp. 445–454 (2020)
22. Heinzerling, B., Strube, M.: Bpemb: Tokenization-free pre-trained subword embeddings in 275 languages. arXiv preprint arXiv:1710.02187 (2017)
23. Hertzum, M., Hornbæk, K.: The effect of target precuing on pointing with mouse and touchpad. International Journal of Human-Computer Interaction **29**(5), 338–350 (2013)
24. Hoppe, S., Loetscher, T., Morey, S.A., Bulling, A.: Eye movements during everyday behavior predict personality traits. Frontiers in human neuroscience p. 105 (2018)
25. Hu, Z., Bulling, A., Li, S., Wang, G.: Fixationnet: Forecasting eye fixations in task-oriented virtual environments. IEEE Transactions on Visualization and Computer Graphics **27**(5), 2681–2690 (2021)
26. Hu, Z., Bulling, A., Li, S., Wang, G.: Ehtask: recognizing user tasks from eye and head movements in immersive virtual reality. IEEE Transactions on Visualization and Computer Graphics (2022)
27. Hu, Z., Li, S., Gai, M.: Research progress of user task prediction and algorithm analysis. Journal of Graphics **42**(3), 367–375 (2021). https://doi.org/http://www.txxb.com.cn/CN/10.11996/JG.j.2095-302X.2021030367
28. Hu, Z., Li, S., Zhang, C., Yi, K., Wang, G., Manocha, D.: Dgaze: Cnn-based gaze prediction in dynamic scenes. IEEE Transactions on Visualization and Computer Graphics **26**(5), 1902–1911 (2020)
29. Hu, Z., Zhang, C., Li, S., Wang, G., Manocha, D.: Sgaze: a data-driven eye-head coordination model for realtime gaze prediction. IEEE Transactions on Visualization and Computer Graphics **25**(5), 2002–2010 (2019)
30. Inoue, H., Hirayama, T., Doman, K., Kawanishi, Y., Ide, I., Deguchi, D., Murase, H.: A classification method of cooking operations based on eye movement patterns.

In: Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications. pp. 205–208 (2016)

31. Jansen, B.J., Jung, S.G., Robillos, D.R., Salminen, J.: Next likely behavior: Predicting individual actions from aggregate user behaviors. In: 2021 Second International Conference on Intelligent Data Science Technologies and Applications (IDSTA). pp. 11–15. IEEE (2021)
32. Jawahar, G., Sagot, B., Seddah, D.: What does bert learn about the structure of language? In: ACL 2019-57th Annual Meeting of the Association for Computational Linguistics (2019)
33. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 133–142 (2002)
34. Kim, Y., Jernite, Y., Sontag, D., Rush, A.M.: Character-aware neural language models. In: Thirtieth AAAI conference on artificial intelligence (2016)
35. Kudo, T.: Subword regularization: Improving neural network translation models with multiple subword candidates. arXiv preprint arXiv:1804.10959 (2018)
36. Kunchukuttan, A., Bhattacharyya, P.: Learning variable length units for smt between related languages via byte pair encoding. arXiv preprint arXiv:1610.06510 (2016)
37. Li, J., Luong, M.T., Jurafsky, D.: A hierarchical neural autoencoder for paragraphs and documents. arXiv preprint arXiv:1506.01057 (2015)
38. Li, T.J.J., Popowski, L., Mitchell, T.M., Myers, B.A.: Screen2vec: Semantic embedding of gui screens and gui components. Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (2021)
39. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
40. Mazilu, S., Blanke, U., Dorfman, M., Gazit, E., Mirelman, A., M. Hausdorff, J., Tröster, G.: A wearable assistant for gait training for parkinson's disease with freezing of gait in out-of-the-lab environments. ACM Transactions on Interactive Intelligent Systems (TiiS) **5**(1), 1–31 (2015)
41. Motwani, A., Jain, R., Sondhi, J.: A multimodal behavioral biometric technique for user identification using mouse and keystroke dynamics. International Journal of Computer Applications **111**(8), 15–20 (2015)
42. Müller, P., Huang, M.X., Bulling, A.: Detecting low rapport during natural interactions in small groups from non-verbal behaviour. In: 23rd International Conference on Intelligent User Interfaces. pp. 153–164 (2018)
43. Murphy, C., Huang, J., Hou, D., Schuckers, S.: Shared dataset on natural human-computer interaction to support continuous authentication research. In: 2017 IEEE International Joint Conference on Biometrics (IJCB). pp. 525–530. IEEE (2017)
44. Pasqual, P.T., Wobbrock, J.O.: Mouse pointing endpoint prediction using kinematic template matching. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 743–752 (2014)
45. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: EMNLP (2014)
46. Petersen, G.B., Mottelson, A., Makransky, G.: Pedagogical agents in educational vr: An in the wild study. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. pp. 1–12 (2021)
47. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)

48. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(140), 1–67 (2020)
49. Reani, M., Peek, N., Jay, C.: An investigation of the effects of n-gram length in scanpath analysis for eye-tracking research. In: Proceedings of the 2018 acm symposium on eye tracking research & applications. pp. 1–8 (2018)
50. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. ArXiv **abs/1908.10084** (2019)
51. Salmeron-Majadas, S., Santos, O.C., Boticario, J.G.: An evaluation of mouse and keyboard interaction indicators towards non-intrusive and low cost affective modeling in an educational context. Procedia Computer Science **35**, 691–700 (2014)
52. Salvucci, D.D., Goldberg, J.H.: Identifying fixations and saccades in eye-tracking protocols. In: Proceedings of the 2000 symposium on Eye tracking research & applications. pp. 71–78 (2000)
53. Schuster, M., Nakajima, K.: Japanese and korean voice search. In: 2012 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 5149–5152. IEEE (2012)
54. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909 (2015)
55. Shirahama, K., Grzegorzek, M.: On the generality of codebook approach for sensor-based human activity recognition. Electronics **6**(2), 44 (2017)
56. Soukoreff, R.W., MacKenzie, I.S.: Towards a standard for pointing device evaluation, perspectives on 27 years of fitts' law research in hci. International Journal of Human-Computer Studies **61**(6), 751–789 (2004)
57. Subba, B., Biswas, S., Karmakar, S.: Host based intrusion detection system using frequency analysis of n-gram terms. In: TENCON 2017-2017 IEEE Region 10 Conference. pp. 2006–2011. IEEE (2017)
58. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune bert for text classification? In: China national conference on Chinese computational linguistics. pp. 194–206. Springer (2019)
59. Sun, Y., Ceker, H., Upadhyaya, S.: Shared keystroke dataset for continuous authentication. In: 2016 IEEE International Workshop on Information Forensics and Security (WIFS). pp. 1–6. IEEE (2016)
60. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is All you Need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
61. Wang, B., Li, G., Zhou, X., Chen, Z., Grossman, T., Li, Y.: Screen2words: Automatic mobile ui summarization with multimodal learning. The 34th Annual ACM Symposium on User Interface Software and Technology (2021)
62. Wang, C., Cho, K., Gu, J.: Neural machine translation with byte-level subwords. In: AAAI (2020)
63. Wang, Z., Li, B.: Human activity encoding and recognition using low-level visual features. In: Twenty-First International Joint Conference on Artificial Intelligence. Citeseer (2009)
64. Wulff, D.U., Haslbeck, J.M., Kieslich, P.J., Henninger, F., Schulte-Mecklenbeck, M.: Mouse-tracking: Detecting types in movement trajectories. In: A Handbook of process tracing methods, pp. 131–145. Routledge (2019)
65. Xiaofeng, L., Shengfei, Z., Shengwei, Y.: Continuous authentication by free-text keystroke based on cnn plus rnn. Procedia computer science **147**, 314–318 (2019)

66. Xu, P., Sugano, Y., Bulling, A.: Spatio-temporal modeling and prediction of visual attention in graphical user interfaces. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. pp. 3299–3310 (2016)
67. Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., Xu, B.: Ts2vec: Towards universal representation of time series. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 8980–8987 (2022)
68. Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., Eickhoff, C.: A transformer-based framework for multivariate time series representation learning. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 2114–2124 (2021)
69. Zhan, J., Liao, X., Bao, Y., Gan, L., Tan, Z., Zhang, M., He, R., Lu, J.: An effective feature representation of web log data by leveraging byte pair encoding and tf-idf. In: Proceedings of the ACM Turing Celebration Conference-China. pp. 1–6 (2019)
70. Zhang, G., Hindennach, S., Leusmann, J., Bühler, F., Steuerlein, B., Mayer, S., Bâce, M., Bulling, A.: Predicting next actions and latent intents during text formatting (2022)
71. Zhao, Y., Miao, D., Cai, Z.: Reading personality preferences from motion patterns in computer mouse operations. IEEE Transactions on Affective Computing (2020)

A Preprocessing Mouse Data with I-DT

As written in Section 4.1, the Dispersion-threshold identification (I-DT) algorithm [52] was used to categorise mouse behaviour to *pinpointing* a target (resembling gaze fixations) and *re-direction* between targets. I-DT operates on a window of duration-threshold consecutive samples. On this window, it calculates the dispersion value as $Dispersion = [max(x) - min(x)] + [max(y) - min(y)]$. If the dispersion value exceeds the dispersion threshold, samples inside the window are not considered to belong to a pinpoint and the window is slid forward by one sample. If the value is below the threshold, the samples within the window are considered to belong to a pinpoint. The window then expands to incorporate new samples until the dispersion value is above the threshold again. We empirically set the duration threshold to 100 ms and the dispersion threshold to 0.1.

B The Algorithm of Byte Pair Encoding

Algorithm 1 shows how byte pair encoding (BPE) constructs the vocabulary V, as introduced in Section 4.2.

C Analysis of EMAKI Data Amount for Interactive Task Recognition

As written in Section 6.3, we evaluated if the size of EMAKI allows our data-driven method to recognise interactive tasks. We used different percentages of the training set to train the method and examined their performances. According to Figure 6c, the best results were achieved by BPE-300 when windows have 150

Algorithm 1 Byte pair encoding (BPE) [69,22]

```
Input: action sequence set $D$, the number of iterations $k$
procedure BPE($D$, $k$)
    $V \leftarrow$ all unique actions in $D$
    for $i \leftarrow 1$ to $k$ do
        $t_L, t_R \leftarrow$ Most frequent two consecutive units (actions or activities) in $D$
        $t_{\text{new}} \leftarrow t_L + t_R$                    ▷ Merge to form a new activity
        $V \leftarrow V + [t_{\text{new}}]$
        Replace each occurrence of $t_L, t_R$ with $t_{\text{new}}$ in $D$
    end for
    return $V$
end procedure
```

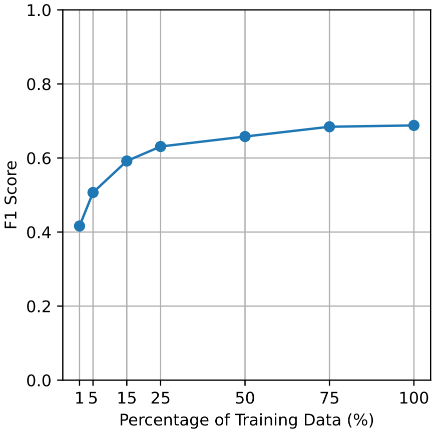


Fig. 9. F1 scores of interactive task recognition achieved by BPE, trained on different percentages of the training set.

actions. Therefore, we followed the above setting. Fig. 9 shows the results of interactive task recognition by training the model with 1%, 5%, 15%, 25%, 50%, 75% of randomly selected training instances, as well as with the entire training set (100%). It can be seen that as the percentage increases, the F1 score first increases fast (before 25%) but then slowly (25% to 75%). The increase in F1 score from using 75% of training data and the entire training set was subtle (only 0.004). Taken together, the amount of data in our dataset is sufficient to perform interactive task recognition.