Benchmarking Predictive and Recommendation Models for Knowledge Work Productivity on the RLKWiC Dataset

Yuuki Tachioka[1]

[1]Denso IT Laboratory, 13F Shintora Yasuda Bldg., 4-3-1 Shimbashi, Minato-ku, Tokyo, Japan

Abstract

Improving the productivity of knowledge workers is a growing challenge in human-centered computing. This paper presents a benchmark suite built on the RLKWiC dataset, which captures rich behavioral logs and contextual information from real-world digital work environments. We define six practical tasks, including context detection, activity classification, and sequential prediction of web domains, event titles, and applications, designed to reflect realistic productivity support scenarios. We evaluated baseline models that incorporate event- and session-level behavior, using classification and sequence modeling techniques. The results demonstrate that modeling fine-grained user interactions yields consistent performance improvements across tasks. The proposed benchmark provides a reproducible foundation for building recommender systems that proactively support human intent, task continuity, and productivity. By releasing standardized tasks and code, the benchmark addresses the current lack of reproducible evaluation on RLKWiC. Beyond methodological contributions, these tasks provide building blocks for HR applications such as workplace analytics, training support, and well-being monitoring.

Keywords

Knowledge Work, Human-Centered Recommender Systems, Productivity Support, Behavioral Prediction, Benchmark Dataset

1. Introduction

In recent years, improving the productivity of knowledge workers (KWs) has become a highly significant issue both socially and economically [1, 2], particularly in the fields of human-centered computing and recommender systems. KWs must access various types of information, and their productivity is influenced by multiple factors such as the working environment, the psychological state, and the efficiency of information access [3, 4, 5]. Among these, quick access to appropriate information and tools is especially critical [6, 7]. In practice, knowledge work often follows certain behavioral patterns, making it possible to anticipate future tasks or required knowledge [8, 9, 10]. More experienced KWs tend to retrieve information more efficiently, select tools more accurately, and switch tasks more fluently. These observations motivate the need for intelligent systems that can support knowledge work by estimating and recommending the next relevant action, tool, or information based on the behavioral history [11, 12, 13].

To build such systems, high-quality datasets that capture real-world KW behavior are essential. However, publicly available datasets with rich semantic annotations that reflect realistic workflows remain scarce [14]. Among the few, BEHACOM [15] and RLKWiC (Real-Life Knowledge Work in Context)[1] [14] are notable. Although BEHACOM primarily records low-level user actions (e.g., keystrokes, mouse movements), RLKWiC organizes higher-level behavioral structures, contexts, sessions, and events, and includes semantic metadata such as file references, web pages, and DBpedia entities[2] [16].

Despite its rich structure, RLKWiC lacks well-defined benchmark tasks and standardized baselines, which limits its accessibility and broader use in reproducible research. To address this gap, we define six practical tasks grounded in RLKWiC's behavioral and semantic annotations:

1. In-context prediction: Classifying whether a session is aligned with the current task context
2. Knowledge work activity (KWA) label prediction: Multi-label classification of the type of knowledge work
3. Entity relevance estimation: Assessing how DBpedia entities relate to the user's session
4. Web domain recommendation: Predicting the next accessed web domain
5. Event title recommendation: Predicting the next window or content title
6. Application recommendation: Predicting the next application to be used

These tasks, ranging from semantic classification to behavioral prediction, are designed to enable intelligent systems to proactively support user intent and reduce cognitive burden. They reflect real-world productivity support scenarios and form the basis of a reproducible benchmark for the RLKWiC dataset.

Although prior work such as BEHACOM [15] and RLKWiC [14] has provided valuable behavioral datasets, they have not established standardized tasks that allow reproducible comparisons between models. Our work fills this gap by aligning the six benchmark tasks with established research trends in context-aware recommender systems [8, 9] and productivity support tools [6, 7]. For example, sequential prediction tasks are directly related to previous studies on next-domain prediction [17], URL auto-completion, and entity-based recommendation [16]. This positioning ensures that the benchmark tasks are not arbitrary derivations from RLKWiC, but validated scenarios grounded in existing literature and practical needs of knowledge work support.

Section 2 provides an overview of the RLKWiC dataset[3], Section 3 defines the six tasks, Section 4 presents the baseline models, and Section 5 reports experimental results.

Related Work and Positioning. To our knowledge, existing studies that have explicitly used the RLKWiC dataset

RecSys in HR'25: The 5th Workshop on Recommender Systems for Human Resources, in conjunction with the 19th ACM Conference on Recommender Systems, September 22–26, 2025, Prague, Czech Republic.

*Corresponding author.

tachioka.yuki@core.d-itlab.co.jp (Y. Tachioka)

0009-0002-0587-2943 (Y. Tachioka)

[1]https://purl.org/RLKWiC

[2]https://purl.org/entity-recommendation-on-rlkwic

[3]Details are found in Section A in the appendix.

are primarily those conducted by its original authors, focusing on data collection [14] and entity recommendation [16]. In the absence of many independent studies on RLKWiC, we position our benchmark within the broader context of research on context-aware recommender systems [8, 9] and productivity support tools [6, 7]. This situates the six benchmark tasks not only as natural extensions of RLKWiC's annotations, but also as representative of validated needs in the field of knowledge work support. We believe that by releasing standardized benchmark tasks and code, our work will facilitate a wider adoption of RLKWiC, enabling future studies to build on a common foundation.

Beyond methodological contributions, the proposed tasks have direct implications for human resource (HR) systems. For example, in-context prediction could support workplace analytics tools that detect interruptions and provide feedback on focus patterns. The prediction of KWA labels could enable automated profiling of employees' work activities to tailor training or learning support [18, 19]. Assessment of the relevance of the entity can be integrated into knowledge management systems to recommend reference materials that are in line with ongoing tasks [20]. The prediction of application and event title can be applied to intelligent launchers or proactive assistants that reduce the cognitive cost of frequent task switching [21, 22, 23]. These scenarios illustrate how benchmark tasks can serve as building blocks for HR applications that aim to improve employee productivity, well-being, and training effectiveness.

2. RLKWiC Database

The RLKWiC dataset captures diverse knowledge-work behaviors with rich semantic annotations. RLKWiC employs a three-layered hierarchical structure to model user behavior: *contexts*, *sessions*, and *events*. In the highest-level layer, a *context* refers to a user-defined unit of work, such as "lectures," "thesis writing," or "trip planning." This explicit management allows analysis of context switches and multitasking. Next, a *session* represents a coherent block of events within a context. Each session is labeled as "in-context" or "out-of-context". In addition, in-context sessions are annotated with one or more of the 12 KWA labels. In the lowest-level layer, an *event* corresponds to a user interaction. Each event is associated with the following features.

1. Event (window) title and URL: These are concatenated into a single text string (e.g., "Quantum Personalplanung" and "chat.openai.com").
2. Active application: The name of the active application used in the session (80 applications in total, e.g., "default browser", "Telegram").
3. Event cause labels: Categorical labels indicating the trigger for event transitions (17 types in total (Table 9)).

3. Benchmark Tasks

This section defines the six benchmark tasks derived from the RLKWiC dataset. Here, we focus on the design and formulation of each task, including their inputs and outputs at the conceptual level. Implementation details such as feature extraction, embeddings, and model architectures are provided separately in Section 4.

3.1. In-context Prediction

In real-world work environments, users often experience interruptions and these out-of-context activities can introduce noise in knowledge work support systems or user behavior analysis. Therefore, estimating whether an event is in context is a critical task. To address this issue, we formulate a binary classification task that determines whether a given session is in context. A session consists of a sequence of events grouped by a temporal window or by explicit user operations.

As shown in Table 7, the proportion of events in context varies between participants in the RLKWiC dataset. For example, participant p6 shows a particularly low in-context ratio, indicating frequent out-of-context behavior. Since tracking start and stop actions were under the participant's control, the observed in-context ratio may be overestimated. Consequently, constructing a robust in-context prediction model is essential as a foundation for task-aware support systems. For the in-context prediction task, the input is a session consisting of a sequence of events, where each event is associated with three types of feature (title/URL, cause, application). The session-level representation of this sequence is then used for classification. The output is a binary label: 1 if the session is considered in context and 0 otherwise.

For consistency with the KWA label prediction task described in Section 3.2, we adopt the same data split strategy based on a five-fold cross-validation. This ensures that the evaluation results are comparable between the two classification tasks.

3.2. KWA Label Prediction

Each in-context session in the RLKWiC dataset is annotated with one or more of the 12 KWA labels listed in Table 8. These labels indicate the type of intellectual work that is carried out during the session, for example, "Information search," "Authoring," or "Networking." While the RLKWiC dataset provides these labels by manually analyzing participants, such labeling is impractical in real-world applications. Therefore, in this study, we define a multilabel classification task to automatically predict which KWA labels apply to a given in-context session. The task is formulated as a multilabel classification problem: for each in-context session, the goal is to predict a binary `on/off` value for each of the 12 KWA labels. Since a single session may be associated with multiple labels, a multiclass setting is not suitable, and a multilabel setting is adopted instead.

There is a strong class imbalance in KWA label distributions: some labels are rare. To address this, we adopt the following experimental settings and evaluation criteria. Since KWA labels are assigned only to in-context sessions, both training and evaluation are limited to these sessions. To mitigate label imbalance, we partition the data using 5-fold cross-validation such that the label distribution is as uniform as possible across folds.

The input features for this task are the same as those described in Section 3.1. The output is a 12-dimensional binary vector that indicates the `on/off` status of each KWA label.

3.3. Relevance Estimation of DBpedia Entities

In the RLKWiC dataset, each work session is annotated with relevance labels that indicate how strongly the session is related to specific DBpedia entities [16]. This annotation connects the session context to external knowledge bases, aiming to enhance context understanding and knowledge-based recommendation. The relevance is expressed using a three-level label:

- Irrelevant (0): The suggested entity has no meaningful connection to the session context.
- Relevant (1): The entity is somewhat related to the session, but does not fully represent its context.
- Representative (2): The entity is closely aligned with the session context and strongly represents the session's main topic.

Bakhshizadeh et al. [16] proposed a method that uses RDF2Vec to generate knowledge graph embeddings of DBpedia entities and match them with user history to estimate relevance scores. In addition, they introduced an online learning approach that dynamically updates these scores based on user feedback. However, the classification performance reported in that study remains limited. Specifically, the F1 score for the binary task of distinguishing Irrelevant vs Relevant+Representative (0 vs. (1,2)) was 0.686, while that for Irrelevant+Relevant vs Representative ((0,1) vs. 2) was 0.444. Compared to a random baseline (F1 = 0.5), the latter result indicates a particularly weak performance in identifying representative entities. In this study, we define the task as a 3-class classification problem (0 vs. 1 vs. 2). Preliminary analysis showed that online learning with sequential updates posed challenges to reliable prediction. Therefore, we instead adopt a cross-validation setup for performance evaluation.

The input to the model is the title of the events along with the candidate DBpedia entities and the output is the relevance label: 0 (Irrelevant), 1 (Relevant), or 2 (Representative). All labeled entity-session pairs in the dataset are included in the evaluation. For consistency and comparability with previous work [16], in addition to evaluating the model as a 3-class classifier, we also report additional binary classification problems: one for 0 vs. (1,2) and another for (0,1) vs. 2.

3.4. Sequential Domain Recommendation

Predicting the next web domain that a user will access based on their behavior history is a key challenge to understanding user intent and providing contextual task support [8, 17]. If we can anticipate the next domain (e.g., `google.com`, `mail.yahoo.com`, `qiita.com`) a user is likely to visit, it provides valuable cues for inferring the type of task (e.g., web search, email checking, document editing) and the underlying goal. In this task, we predict the next web domain to be accessed on the basis of a user's chronological behavior log. We focus on domains that appear at least three times in the dataset, resulting in a total of 376 unique domains as prediction targets. Practical applications of this task include automatic URL autocompletion tailored to current tasks, intent inference in the early stages, and dynamic presentation of bookmarks or search help. The input consists of a user's recent sequence of domain-level interactions (i.e., accessed domains per event). The output is the next domain predicted to be accessed.

For each user, the behavioral history is sorted in chronological order and split into training, validation, and test sets using a ratio of `0.8:0.1:0.1`. For evaluation, we adopt common ranking metrics such as Hit Rate (Hit@k), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG), which are widely used in recommendation tasks to assess top-k ranked outputs. The data split method and metrics will also be used consistently in subsequent recommendation tasks.

3.5. Sequential Event Title Recommendation

Compared to web domains, event (window) titles offer a more fine-grained signal of user activity, as they often contain explicit information such as search queries, document titles, or visited page contents. Thus, accurate prediction of the next event title can enable a more precise inference of user intent and cognitive state. In this task, we predict the next event title to appear in a user's session stream. We focus on titles that occur at least three times in the RLKWiC dataset, resulting in a total of 2,651 unique titles as prediction candidates. Potential applications of this task include prediction of the next page or query, automatic display of related documents, and reminder prompts during task switching. The input is a chronologically ordered sequence of titles from past events, and the output is the title predicted to occur next.

3.6. Sequential Application Recommendation

Users frequently switch between multiple applications to complete their tasks. For instance, a programmer may refer to API documentation in a Web browser while coding, or a writer may alternate between editing documents and communicating via chat tools. If such application switches can be predicted, it becomes possible to proactively assist users based on their task intent. In this task, we predict the next application that a user will use, focusing on 64 applications that appear at least three times in the dataset. The `cSpaces` application, which is used solely for tracking purposes, is excluded from the prediction targets. Potential applications of this task include intelligent shortcut management, such as dynamically reordering application launch icons or suggesting a swap-style launcher, thus reducing the effort required for application switching. The input is a chronologically ordered sequence of applications used and the output is the next application predicted to be launched or used.

4. Baseline Methods

For each of the benchmark tasks proposed in Section 3, we construct reasonable and comparable baseline methods. The implementation of all baseline models and evaluation scripts will be made publicly available via a GitHub repository[4]. For the *in-context prediction* and *KWA label prediction* tasks described in Sections 3.1 and 3.2, we design Transformer-based classification models that utilize event-level embeddings, as detailed in Sections 4.1 and 4.2. For the *relevance estimation of DBpedia entities* task discussed in Section 3.3, we propose

[4] https://github.com/DensoITLab/RLKWiC_benchmark

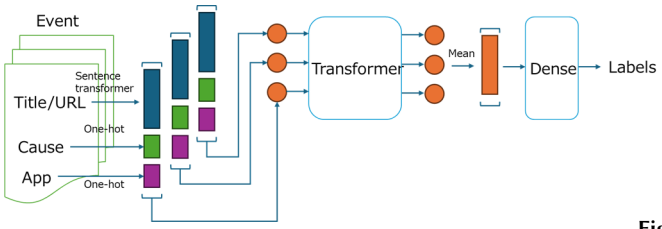

Figure 1: Architecture of the event embedding and sequence classification model.

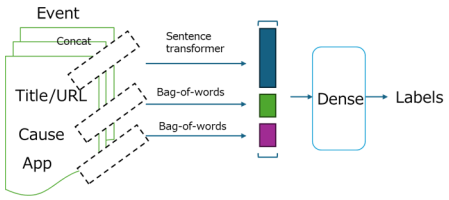

Figure 2: Architecture of the simple session-based classification model.

a model that constructs a contextual representation from the preceding sequences of events and estimates the relevance of the entity via similarity with the corresponding entity vector, as described in Section 4.3.

We used the all-MiniLM-L6-v2 variant of Sentence-BERT [24] for all embedding steps. When the concatenated title and URL string exceeded the model's maximum length, we truncated the input while retaining the most informative segments (title and domain). Cosine similarity scores were assigned to categories 0, 1, or 2 using a pairwise classification layer trained on labeled examples, rather than applying a fixed threshold.

For three types of sequential recommendation tasks composed of *domain*, *event title*, and *application* prediction presented in Sections 3.4, 3.5, and 3.6, we build datasets conforming to the `atomic file` format used in the `RecBole` framework [25], and perform comparative evaluations using representative sequential recommendation models such as GRU4Rec, SASRec, and BERT4Rec. More details are provided in Section 4.4.

4.1. Event Embedding and Sequence Classification Model

In the RLKWiC dataset, the fundamental unit of user behavior is defined as an *event*, each of which is associated with the following attribute information: title/URL, cause label, and active application. Based on this event-level information, we design a Transformer-based session classification model. The overall architecture is illustrated in Figure 1. The input consists of three components:

- Title and URL: Concatenated and embedded into a 384-dimensional vector using Sentence-BERT [24] (specifically, the `all-MiniLM-L6-v2` model).
- Cause label and active application: Both are one-hot encoded and passed through separate fully connected (dense) layers to obtain 16-dimensional dense vectors.

These components are concatenated to form a unified embedding for each event. The sequence of event embeddings is then fed into a Transformer Encoder in chronological order. The final session representation is obtained by averaging the hidden states across all events in the sequence. At the output layer, the aggregated session representation is passed through a fully connected layer to perform either: Binary classification for *in-context prediction*, or Multi-label classification for *KWA label prediction*. This model captures short-term user intent and interest from event sequences, integrates them into a session-level representation via the Transformer, and makes task-specific predictions based on this session embedding.

4.2. Session Embedding and Simple Classification Model

An alternative approach to event-wise modeling is to treat an entire session as a single input unit and classify it using static features. In this section, we introduce a simple session-based classification model following this principle. The overall architecture is illustrated in Figure 2. The input to the model consists of three types of features:

- Title and URL: All title and URL strings within a session are concatenated in chronological order and embedded in a 384-dimensional vector using Sentence-BERT [24].
- Cause labels and application types: Rather than using one-hot encodings, we adopt a bag-of-words (BoW) representation that counts the frequency of each label or application type occurring within the session. This results in fixed-length vectors that are independent of the number of events in the session.

These feature vectors (title embedding, cause BoW, and application BoW) are concatenated and fed into a fully connected (dense) layer to predict the target label. This simple feedforward model does not explicitly consider the order or structure of events but instead relies on aggregated statistical features at the session level. Although this model has the advantage of architectural simplicity and efficient training, its inability to model sequential dependencies may limit its performance compared to the Transformer-based event-wise model introduced in Section 4.1, but it may serve as a practical baseline in settings with limited data or computational resources.

4.3. Event-based Relevance Estimation Model

In this section, we propose an embedding-based model for estimating the semantic relevance between a user's event history and a candidate DBpedia entity. The overall architecture is illustrated in Figure 3. Given a pair consisting of a user's event history and a DBpedia entity to be evaluated, the goal is to predict how strongly the entity relates to the user's current context (as defined in Section 3.3). The input to the model is the sequence of events that occurred immediately before the current session. From each event, the *title* text is extracted and embedded in a 384-dimensional vector using Sentence-BERT. The embeddings from multiple events are then averaged to generate a fixed-length vector that represents the user's contextual intent. In parallel, the abstract associated with the candidate DBpedia entity is retrieved and embedded using the same Sentence-BERT model, resulting in an entity representation vector. The similarity between the user context vector and the entity

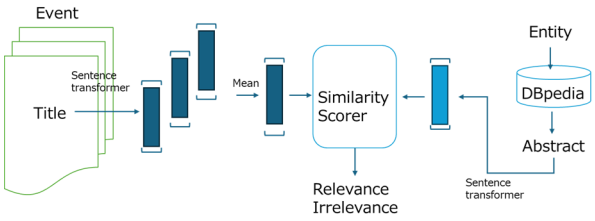

Figure 3: Architecture of the relevance estimation model.

vector is then calculated using cosine similarity[5]. The model estimates the relevance level based on the similarity score, assigning one of three labels: 0 (Irrelevant), 1 (Relevant), or 2 (Representative). The cosine similarity scores are not mapped to categories by fixed thresholds, but we employ a supervised pairwise classification layer that takes the similarity between two vectors as input and predicts one of the three labels. This ensures that the mapping from similarity values to discrete categories is learned from the annotated data rather than predefined heuristics.

4.4. Sequential Recommendation Model

For the three sequential recommendation tasks defined in Sections 3.4 through 3.6, we construct baseline models using a variety of established sequential recommendation methods. To support these tasks, we provide scripts that automatically generate datasets in the `atomic file` format [6] used by the `RecBole` framework [25], comprising `.user`, `.item`, and `.inter` files. This setup enables ranking-based evaluation that utilizes user-level behavioral histories. As shown in Table 10, we evaluated six representative models implemented in RecBole in consistent settings for all three tasks. These models span a diverse range of architectures, including RNNs, CNNs, Transformers, and attention-based mechanisms, enabling comparisons of different sequence modeling strategies [7]. This benchmark allows us to quantify the difficulty of behavioral prediction in the context of knowledge work support, as well as to measure performance differences across model types. Furthermore, comparing architectures provides insight into model design choices and the effectiveness of different feature representations.

5. Results and Discussions

5.1. In-context Prediction Results

Table 1 (event-based model) shows that the event-based model achieved an average F1 score of 75.8% in a five-fold cross-validation[8], with a good balance between accuracy and recall. The model also demonstrated stable performance within the 95% confidence interval. These results suggest that the model architecture, which sequentially incorporates event-level information, is effective in predicting whether a session is in context. This result shows that determining whether sessions are in context from limited features remains a non-trivial task.

Table 1 (session-based model) shows the results of the session-based model, which produced an average F1 score of 68.5%, approximately 7 percentage points lower than that of the event-based model. This performance gap can be attributed to the session-based model's inability to explicitly model the sequential structure of events, relying instead on aggregated features such as BoW and average embeddings. Consequently, it may fail to capture dynamic behavioral changes within a session. Although the event-based model exhibited superior accuracy, the session-based model was more efficient in terms of computation. Specifically, the training and evaluation time per epoch was approximately 3.6 seconds for the session-based model, compared to 25.7 seconds for the event-based model, resulting in a roughly 7x speed-up.

5.2. KWA Label Prediction Results

Table 2 (event-based model) shows that the event-based model achieved an average F1 score of 55.3%. Considering that this is a multi-label classification task with 12 classes and substantial class imbalance, the results suggest that the model has achieved a reasonable level of accuracy. However, recall (0.6227) is relatively higher than precision (0.5577) and Jaccard score (0.4309), suggesting that the model tends to overpredict some labels, leading to less precise but more inclusive predictions.

Table 2 (session-based model) shows the results for the session-based model. In particular, it achieved a slightly higher average F1 score of 56.1%, and the average recall reached 79.3%, which is a substantial improvement over the event-based model. This suggests that the session-based model is more effective at capturing label co-occurrence tendencies across the session, possibly due to the use of BoW representations. As a result, it tends to reduce label omissions and achieve higher recall.

5.3. Entity Relevance Prediction Results

Table 3 summarizes the performance of the proposed model in five-fold cross-validation under three classification settings: two binary classification setups and one three-class classification.

In the first setting, we distinguish irrelevant entities (label = 0) from the rest (labels = 1 or 2). The model achieved a high average F1 score of 81.9%. This indicates a strong semantic similarity between the user context derived from the event history and the knowledge embedding derived from the entity abstract. These results validate the effectiveness of sentence transformer-based representations and cosine similarity scoring.

In the second setting, we isolate representative entities (label = 2) from the other two categories (labels = 0 and 1). Here, the model achieved an even higher average F1 score of 85.9%. This suggests that entities labeled as "Representative" form semantically distinct clusters in the embedding space and that the model effectively captures this distinction. Compared to previous work [16], where this binary split yielded lower accuracy, our results indicate that identifying "representativeness" can be learned as a meaningful evaluation metric.

In the more challenging three-class classification setting, the model still achieved a solid average F1 score of 67.6%. However, the relatively wide 95% confidence intervals sug-

[5]Alternatively, a pairwise classification model that directly takes the two vectors as input and predicts the relevance class can also be considered.
[6]https://recbole.io/docs/user_guide/data/atomic_files.html
[7]The details of selected model architectures are shown in Appendix-B.1.
[8]Further fold-wise results and implementation details are provided in Appendix-B.2.

Table 1
Results for in-context prediction (event-based and session-based model).

Model		Accuracy	Precision	Recall	F1-Score
event-based model	Mean	0.7872	0.7694	0.7506	0.7576
	95% CI	[0.7606, 0.8137]	[0.7379, 0.8008]	[0.7280, 0.7732]	[0.7352, 0.7800]
session-based model	Mean	0.7183	0.6894	0.6863	0.6854
	95% CI	[0.6706, 0.7659]	[0.6481, 0.7306]	[0.6284, 0.7442]	[0.6325, 0.7382]

Table 2
Results for KWA label prediction (event-based and session-based model).

Model		Jaccard Score	Precision	Recall	F1-Score
event-based model	Mean	0.4309	0.5577	0.6227	0.5534
	95% CI	[0.3993, 0.4625]	[0.5221, 0.5933]	[0.5642, 0.6813]	[0.5161, 0.5906]
session-based model	Mean	0.4158	0.5280	0.7925	0.5612
	95% CI	[0.3394, 0.4922]	[0.4250, 0.6310]	[0.6566, 0.9284]	[0.4900, 0.6324]

gest variability across folds, likely due to contextual ambiguity or subjective differences in user annotations.

5.4. Sequential Web Domain Recommendation Results

Table 4 shows the results of the web domain prediction task (defined in Section 3.4). We compared six sequential recommendation models implemented in `RecBole`. In general, NextItNet achieved the highest prediction performance in most metrics, including Hit@1 (0.1783), MRR, and NDCG. Furthermore, NARM outperformed all other models in terms of Hit@5 and NDCG@5, making it another strong candidate among baselines. In contrast, SASRec, BERT4Rec, and GRU4Rec demonstrated relatively lower performance. SASRec achieved a Hit@1 of only 0.0864, indicating potential limitations in its ability to capture contextual signals from short-term histories. NextItNet's architecture, which employs dilated convolutions to model long-range dependencies efficiently, appears particularly well-suited for this task. Its ability to explicitly and hierarchically represent broader contexts suggests the effectiveness of CNN-based models in domain-level prediction. Similarly, NARM integrates an attention mechanism into a GRU-based sequence model, allowing it to dynamically combine both short- and long-term user intent for improved recommendation quality. However, self-attention-based models, such as SASRec and BERT4Rec, may be less aligned with the characteristics of this task. Since web domains are relatively abstract and categorical compared to concrete item IDs or page titles, global contextual patterns may be more important than local sequential dependencies, potentially explaining their underperformance in this setting.

5.5. Sequential Event Title Recommendation Results

Table 5 shows the results for the event title prediction task (defined in Section 3.5). Among all models, NextItNet achieved the highest performance, outperforming others in terms of Hit@1 (0.0808), MRR, and NDCG. These results suggest that CNN-based architectures, which can efficiently capture long-range dependencies in sequential data, are also effective for next-step event-level predictions. SASRec and NARM were close followers, and SASRec achieved the best performance in terms of Hit@5, indicating that its self-attention mechanism is capable of dynamically attending to contextually important events in the input history. In contrast, GRU4Rec showed the lowest performance in all metrics. This result highlights the limitations of simple RNNs in handling highly diverse and semantically rich output spaces such as event titles.

5.6. Sequential Application Recommendation Results

Table 6 shows the results of the application prediction task (defined in Section 3.6). In this task, all models demonstrated relatively high performance. NARM, NextItNet, and GRU4Rec emerged as the top performers. Application prediction likely depends on the short-term context, and NARM scored the best on all metrics except Hit@1, indicating its strong effectiveness for this task. NextItNet also consistently ranked high in all metrics. GRU4Rec achieved the best Hit@1 score (0.6040). Importantly, all three leading models (NARM, NextItNet, and GRU4Rec) achieved more than 85% in Hit@5, indicating practical feasibility for application-switching support systems. For example, the top five predicted applications could be presented as shortcut buttons, significantly reducing the user's switching effort. Application usage is closely tied to user tasks and workflow structure, and patterns are often stable. Therefore, sequential models are particularly well suited to this task.

5.7. Limitation

Despite its contribution, our work has several limitations. First, the RLKWiC dataset was collected from only eight university students in Germany [14], which restricts the demographic and occupational diversity of the sample. As a result, the generalizability of the reported benchmark performance remains limited. Second, participants had control over recording start and stop actions, which may have led to biases in the proportion of in-context versus out-of-context sessions. Third, the dataset exhibits strong label imbalance, especially for rare knowledge work activities (Table 8) and representative DBpedia entities (Table 3), which complicates model training. Finally, the relatively wide 95% confidence intervals observed in several tasks (e.g., in-context prediction and entity relevance estimation) indicate vari-

Table 3
Five-fold cross-validation results for the DBpedia entity relevance prediction task.

Recommendation Task		Accuracy	Precision	Recall	F1-Score
0 vs. (1,2) discrimination	Mean	0.8293	0.8277	0.8184	0.8191
	95% CI	[0.7886, 0.8701]	[0.7885, 0.8669]	[0.7791, 0.8578]	[0.7804, 0.8577]
(0,1) vs. 2 discrimination	Mean	0.8827	0.8661	0.8578	0.8591
	95% CI	[0.8377, 0.9277]	[0.8103, 0.9219]	[0.8051, 0.9105]	[0.8102, 0.9080]
0 vs. 1 vs. 2 discrimination	Mean	0.7096	0.6887	0.6829	0.6757
	95% CI	[0.6395, 0.7796]	[0.6132, 0.7642]	[0.6125, 0.7533]	[0.5991, 0.7523]

Table 4
Results for the sequential domain recommendation task.

Method	Hit@1	Hit@5	Hit@10	MRR@5	MRR@10	NDCG@5	NDCG@10
SASRec	0.0864	0.2629	0.3585	0.1378	0.1512	0.1681	0.1997
BERT4Rec	0.0772	0.3474	0.5055	0.1839	0.2060	0.2250	0.2771
GRU4Rec	0.0901	0.2739	0.3805	0.1439	0.1566	0.1756	0.2086
FPMC	0.1213	0.3511	0.5257	0.1867	0.2106	0.2264	0.2835
NextItNet	**0.1783**	0.4191	**0.5570**	**0.2575**	**0.2767**	0.2972	**0.3425**
NARM	0.1544	**0.4393**	0.5496	0.2536	0.2685	**0.2995**	0.3353

ance across folds and highlight the need for larger-scale datasets. Future work should extend evaluations to cross-user splits, where the full data of certain participants is held out, to better assess the generalization capability of predictive models.

6. Conclusion

This study presents a practical approach to building a benchmark suite designed to support knowledge work. Leveraging RLKWiC's rich semantics and multilayered structure, we defined the following six benchmark tasks: (1) In-context prediction, (2) KWA label classification, (3) Relevance estimation with DBpedia entities, (4) Web domain prediction, (5) Event title prediction, and (6) Application prediction. For each task, we proposed appropriate baseline models: event/session-level embedding-based classifiers, a relevance estimation model for entity matching, and sequential recommendation models.

The results indicate that Transformer-based models operating on event sequences achieved strong performance for in-context detection and KWA classification. Sentence embedding-based similarity scoring proved effective for relevance estimation. Sequential models such as NextItNet and NARM achieved high accuracy in predicting domains, events, and applications.

These benchmark tasks and their results provide a solid foundation for future research in knowledge work support and behavioral prediction systems. We hope that this work serves as a standard reference point.

Declaration on Generative AI

During the preparation of this work, the author used ChatGPT-4 and writefull in order to: Grammar and spelling check. After using these services, the author reviewed and

Table 5
Results for the sequential event title recommendation task.

Method	Hit@1	Hit@5	Hit@10	MRR@5	MRR@10	NDCG@5	NDCG@10
SASRec	0.0654	0.1950	0.2438	0.1136	0.1201	0.1340	0.1497
BERT4Rec	0.0664	0.1645	0.2237	0.1002	0.1083	0.1161	0.1354
GRU4Rec	0.0467	0.1038	0.1271	0.0662	0.0692	0.0755	0.0830
FPMC	0.0578	0.1799	**0.2535**	0.1012	0.1107	0.1207	0.1442
NextItNet	**0.0808**	**0.1978**	0.2503	**0.1247**	**0.1318**	**0.1429**	**0.1601**
NARM	0.0643	0.1899	0.2521	0.1105	0.1189	0.1303	0.1505

Table 6
Results for the sequential application recommendation task.

Method	Hit@1	Hit@5	Hit@10	Recall@5	Recall@10	NDCG@5	NDCG@10
SASRec	0.5692	0.8553	0.9396	0.6755	0.6870	0.7204	0.7478
BERT4Rec	0.5714	0.8421	0.9303	0.6678	0.6801	0.7110	0.7401
GRU4Rec	0.6040	0.8654	0.9411	0.7017	0.7120	0.7425	0.7673
FPMC	0.5332	0.7623	0.8916	0.6149	0.6321	0.6515	0.6933
NextItNet	**0.6164**	0.8609	0.9408	0.7092	0.7198	0.7471	0.7729
NARM	0.6097	**0.8811**	**0.9501**	**0.7112**	**0.7206**	**0.7536**	**0.7761**

edited the content as needed and takes full responsibility for the publication's content.

References

[1] T. H. Davenport, Thinking for a living: How to get better performance and results from knowledge workers, Harvard Business Press, Brighton, Massachusetts, USA, 2005.

[2] P. F. Drucker, Management Challenges for the 21st Century, HarperBusiness, New York, 1999.

[3] M. Palvalin, What matters for knowledge work productivity?, Employee Relations 41 (2019) 209–227. URL: https://doi.org/10.1108/ER-04-2017-0091. doi:10.1108/ER-04-2017-0091.

[4] M. Züger, C. Corley, A. N. Meyer, B. Li, T. Fritz, D. Shepherd, V. Augustine, P. Francis, N. Kraft, W. Snipes, Reducing interruptions at work: A large-scale field study of flowlight, in: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 61–72. URL: https://doi.org/10.1145/3025453.3025662. doi:10.1145/3025453.3025662.

[5] G. Mark, D. Gudith, U. Klocke, The cost of interrupted work: more speed and stress, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '08, Association for Computing Machinery, New York, NY, USA, 2008, pp. 107–110. URL: https://doi.org/10.1145/1357054.1357072. doi:10.1145/1357054.1357072.

[6] S. Geng, L. Tan, B. Niu, Y. Feng, L. Chen, Knowledge recommendation for workplace learning: a system design and evaluation perspective, Internet Research 30 (2020) 243–261. URL: https://doi.org/10.1108/INTR-07-2018-0336. doi:10.1108/INTR-07-2018-0336.

[7] E. Horvitz, C. Kadie, T. Paek, D. Hovel, Models of attention in computing and communication: from principles to applications, Commun. ACM 46 (2003) 52–59. URL: https://doi.org/10.1145/636772.636798. doi:10.1145/636772.636798.

[8] G. Adomavicius, B. Mobasher, F. Ricci, A. Tuzhilin, Context-aware recommender systems, AI Magazine 32 (2011) 67–80. URL: https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/2364. doi:10.1609/aimag.v32i3.2364.

[9] P. Mateos, A. Bellogin, A systematic literature review of recent advances on context-aware recommender systems, Artificial Intelligence Review 58 (2024) 20. URL: https://doi.org/10.1007/s10462-024-10939-4. doi:10.1007/s10462-024-10939-4.

[10] J. Teevan, S. T. Dumais, E. Horvitz, Personalizing search via automated analysis of interests and activities, in: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '05, Association for Computing Machinery, New York, NY, USA, 2005, pp. 449–456. URL: https://doi.org/10.1145/1076034.1076111. doi:10.1145/1076034.1076111.

[11] R. Chamun, D. Pinheiro, D. Jornada, J. a. B. S. de Oliveira, I. Manssour, Extracting web content for personalized presentation, in: Proceedings of the 2014 ACM Symposium on Document Engineering, DocEng '14, Association for Computing Machinery, New York, NY, USA, 2014, pp. 157–164. URL: https://doi.org/10.1145/2644866.2644871. doi:10.1145/2644866.2644871.

[12] K. Tanaka, Web knowledge extraction for improving search, in: Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication, ICUIMC '08, Association for Computing Machinery, New York, NY, USA, 2008, pp. 140–145. URL: https://doi.org/10.1145/1352793.1352823. doi:10.1145/1352793.1352823.

[13] M. M. Kamel, A. Gil-Solla, M. Ramos-Carber, Tasks recommendation in crowdsourcing based on workers' implicit profiles and performance history, in: Proceedings of the 9th International Conference on Software and Information Engineering, ICSIE '20, Association for Computing Machinery, New York, NY, USA, 2021, pp. 51–55. URL: https://doi.org/10.1145/3436829.3436834. doi:10.1145/3436829.3436834.

[14] M. Bakhshizadeh, C. Jilek, M. Schröder, H. Maus, A. Dengel, Data collection of real-life knowledge work in context: The RLKWiC dataset, in: S. Li (Ed.), Information Management, Springer Nature Switzerland, Cham, 2024, pp. 277–290.

[15] P. M. Sanchez Sanchez, J. M. Jorquera Valero, M. Zago, A. Huertas Celdran, L. Fernandez Maimo, E. Lopez Bernal, S. Lopez Bernal, J. Martinez Valverde, P. Nespoli, J. Pastor Galindo, Angel L. Perales Gomez, M. Gil Perez, G. Martinez Perez, BEHACOM - a dataset modelling users' behaviour in computers, Data in Brief 31 (2020) 105767. URL: https://www.sciencedirect.com/science/article/pii/S2352340920306612. doi:https://doi.org/10.1016/j.dib.2020.105767.

[16] M. Bakhshizadeh, H. Maus, A. Dengel, Context-based entity recommendation for knowledge workers: Establishing a benchmark on real-life data, in: Proceedings of the 18th ACM Conference on Recommender Systems, RecSys '24, Association for Computing Machinery, New York, NY, USA, 2024, pp. 654–659. URL: https://doi.org/10.1145/3640457.3688068. doi:10.1145/3640457.3688068.

[17] W. Hussein, T. F. Gharib, R. M. Ismail, M. G.-H. M. Mostafa, A user-concept matrix clustering algorithm for efficient next page prediction, Int. J. Knowl. Web Intell. 5 (2016) 208–229. URL: https://doi.org/10.1504/IJKWI.2016.078718. doi:10.1504/IJKWI.2016.078718.

[18] M. Granitzer, A. S. Rath, M. Kröll, C. Seifert, D. Ipsmiller, D. Devaurs, N. Weber, S. Lindstaedt, Machine learning based work task classification, Journal of Digital Information Management 7 (2009) 306–313. URL: https://hal.science/hal-00872101.

[19] R. L. Jacobs, Work Analysis in the Knowledge Economy, Palgrave Macmillan, 2019. doi:10.1007/978-3-319-94448-7.

[20] G. Jacucci, P. Daee, T. Vuong, S. Andolina, K. Klouche, M. Sjöberg, T. Ruotsalo, S. Kaski, Entity recommendation for everyday digital tasks, ACM Trans. Comput.-Hum. Interact. 28 (2021). URL: https://doi.org/10.1145/3458919. doi:10.1145/3458919.

[21] M. Kersten, G. C. Murphy, Reducing friction for knowledge workers with task context, AI Mag. 36 (2015) 33–41. URL: https://doi.org/10.1609/aimag.v36i2.2581. doi:10.1609/aimag.v36i2.2581.

[22] M. Aliannejadi, H. Zamani, F. Crestani, W. B. Croft, Context-aware target apps selection and recommen-

dation for enhancing personal mobile assistants, ACM Trans. Inf. Syst. 39 (2021). URL: https://doi.org/10.1145/3447678. doi:10.1145/3447678.

[23] M. Chis, H. Li, K. Zheng, M. Lewis, D. Hughes, K. Sycara, The cognitive load – productivity tradeoff in task switching, Proceedings of the Human Factors and Ergonomics Society Annual Meeting 67 (2023) 666–671. URL: https://doi.org/10.1177/21695067231193677. doi:10.1177/21695067231193677.

[24] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. URL: https://aclanthology.org/D19-1410/. doi:10.18653/v1/D19-1410.

[25] W. X. Zhao, Y. Hou, X. Pan, C. Yang, Z. Zhang, Z. Lin, J. Zhang, S. Bian, J. Tang, W. Sun, Y. Chen, L. Xu, G. Zhang, Z. Tian, C. Tian, S. Mu, X. Fan, X. Chen, J.-R. Wen, RecBole 2.0: Towards a more up-to-date recommendation library, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, CIKM '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 4722–4726. URL: https://doi.org/10.1145/3511808.3557680. doi:10.1145/3511808.3557680.

[26] W.-C. Kang, J. McAuley, Self-attentive sequential recommendation, in: 2018 IEEE International Conference on Data Mining (ICDM), IEEE Computer Society, Los Alamitos, CA, USA, 2018, pp. 197–206. URL: https://doi.ieeecomputersociety.org/10.1109/ICDM.2018.00035. doi:10.1109/ICDM.2018.00035.

[27] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, P. Jiang, BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1441–1450. URL: https://doi.org/10.1145/3357384.3357895. doi:10.1145/3357384.3357895.

[28] B. Hidasi, A. Karatzoglou, L. Baltrunas, D. Tikk, Session-based recommendations with recurrent neural networks, arXiv preprint arXiv:1511.06939, 2016. URL: https://arxiv.org/abs/1511.06939. arXiv:1511.06939.

[29] S. Rendle, C. Freudenthaler, L. Schmidt-Thieme, Factorizing personalized Markov chains for next-basket recommendation, in: Proceedings of the 19th International Conference on World Wide Web, WWW '10, Association for Computing Machinery, New York, NY, USA, 2010, pp. 811–820. URL: https://doi.org/10.1145/1772690.1772773. doi:10.1145/1772690.1772773.

[30] F. Yuan, A. Karatzoglou, I. Arapakis, J. M. Jose, X. He, A simple convolutional generative network for next item recommendation, in: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 582–590. URL: https://doi.org/10.1145/3289600.3290975. doi:10.1145/3289600.3290975.

[31] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, J. Ma, Neural attentive session-based recommendation, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 1419–1428. URL: https://doi.org/10.1145/3132847.3132926. doi:10.1145/3132847.3132926.

Appendix

A. Details of RLKWiC Database

The RLKWiC dataset is a highly valuable resource that captures diverse knowledge-work behaviors in real-world knowledge-work environments with rich semantic annotations. The dataset was collected over approximately two months, from the end of May to early July 2023, from eight students (aged 23 to 35) at the University of Kaiserslautern-Landau in Germany. Data collection was carried out using two tools: cSpaces, which allows participants to explicitly manage their work contexts, and the User Activity Tracker, which automatically records user interaction logs. As a result, a wide range of information was recorded in detail, including active window titles, applications used, clipboard contents, file operations, browsing history, and context switches. For privacy protection, participants had full control over recording, deleting collected data, and applying anonymization.

Table 7 provides aggregated statistics for the RLKWiC dataset, showing the number of contexts, sessions, events, total durations, and in-context ratios per participant. These figures illustrate the variation in working styles and context-tracking practices. For example, participant p6 has a significantly lower in-context ratio (25.2%), suggesting frequent interruptions or less precise context labeling, while others such as p5 and p7 show very high in-context ratios above 95%. The table also highlights that all but two participants recorded more than 10,000 minutes of events, ensuring sufficient data volume for analysis.

RLKWiC employs a three-layered hierarchical structure to model user behavior: *contexts*, *sessions*, and *events*. In the highest-level layer, a *context* refers to a user-defined unit of work, such as "lectures," "thesis writing," or "trip planning." Through cSpaces, users could flexibly create new contexts or switch between existing ones depending on their current task. This explicit management enables analysis of context switches and multitasking. Next, a *session* represents a coherent block of events within a context. Each session is labeled as "in-context" or "out-of-context". In-context sessions are defined as sessions are semantically aligned with the user's self-declared current task or objective. In contrast, out-of-context sessions include unrelated or interruptive sessions, such as administrative operations or personal browsing, that are not directly tied to the ongoing work context.

In-context sessions are further annotated with one or more Knowledge Work Activity (KWA) labels. Table 8 lists the 12 KWA categories (e.g., "Information search," "Learning," "Authoring") and their frequency across the dataset. This annotation enables task-level analysis such as focus distribution and label co-occurrence across sessions. It also serves as the target for the KWA label classification tasks in Section 3.2.

Table 7
Statistics of the RLKWiC dataset: context-aware and total sessions, events, and durations per participant.

ID	context	session		event		minutes (day)		in-context
		in-context	total	in-context	total	in-context	total	ratio (%)
p1	11	106	152	14,441	16,506	15,115 (10.5)	17,207 (12.0)	87.9
p2	4	11	15	385	414	483 (0.3)	518 (0.4)	93.2
p3	4	75	121	8,310	11,869	23,742 (16.5)	29,252 (20.3)	81.2
p4	4	51	60	4,878	5,404	5,265 (3.7)	5,973 (4.1)	88.2
p5	3	13	18	1,132	1,428	14,088 (9.8)	14,450 (10.0)	97.5
p6	10	55	142	7,492	14,280	8,310 (5.8)	32,919 (22.9)	25.2
p7	8	40	55	1,613	2,181	14,931 (10.4)	15,592 (10.8)	95.8
p8	11	74	90	7,610	8,765	9,439 (6.6)	9,887 (6.9)	95.5
Total	**55**	**425**	**653**	**45,761**	**61,247**	**91,373 (63.5)**	**125,798 (87.4)**	**72.6**

Table 8
Knowledge Work Activity (KWA) labels and their number of appearances in the RLKWiC dataset.

Label	#	Label	#	Label	#	Label	#
Acquisition	213	Information organization	173	Authoring	94	Dissemination	44
Information search	212	Analyze	142	Expert search	88	Service search	44
Learning	195	Networking	110	Feedback	58	Monitoring	44

In the lowest-level layer, an *event* corresponds to a user interaction with timestamp, such as application launches, window switches, file operations, or clipboard actions. These fine-grained logs are crucial for mining behavior patterns, estimating user focus, and building predictive interaction models. Each event is associated with the following features.

1. Event (window) title and URL: These are concatenated into a single text string.
2. Active application: The name of the active application used in the session (80 applications in total).
3. Event cause labels: Categorical labels indicating the trigger for event transitions (17 types in total), as detailed in Table 9.

Table 9 summarizes all cause labels recorded in the dataset and their frequency. The most frequent cause is "active window changed" with more than 42,000 occurrences, reflecting the application or window switch behavior. Other notable causes include web visits (focused or visible), context switches, file drops, and tagging operations. These categorical labels provide rich signals to understand user intent and trigger conditions in multi-tasking environments.

In addition to the hierarchical structure, RLKWiC includes metadata on the documents accessed by users. For local files and web pages, metadata such as filenames, file paths, visited URLs, page titles, and access timestamps are recorded and linked to the corresponding context. This enables a comprehensive analysis of information-seeking behavior and reference history. Furthermore, the dataset is enriched with lexical and semantic features. It includes bag-of-words and stemmed tokens extracted from documents and webpages, as well as entity links to DBpedia. This allows documents to be associated with concepts such as organizations, locations, or academic topics, facilitating semantic search, knowledge graph construction, and entity-based recommendations [16]. In summary, RLKWiC is a uniquely comprehensive dataset that integrates layered information on work contexts, behavior logs, reference materials, and semantic structures, offering a solid foundation for analysis and support of knowledge work.

Table 9
Event cause labels and their number of appearances in the RLKWiC dataset. These labels represent the cause of event transitions.

Label	#	Label	#	Label	#
active window changed	42,701	an item was removed from the context's activity history	407	new context was created	58
a webpage was visited (window focus changed)	8,682	a folder was rebirthed to a context (by adding tags)	281	more context's activity history was browsed	57
a webpage was visited (visibility changed)	7,545	new item was added to the context	281	an item from the context's activity history was opened	55
search keywords were entered	855	new tag was added	226	an item was removed from the context	53
the selected context was switched	664	a file was dropped on the cSpaces sidebar	113	tag was removed from the context	14
observation switched on/off	464	an item from the context was opened	95		

Table 10
Sequential recommendation models used in the experiment.

Model	Architecture Type	Key Characteristics
SASRec [26]	Transformer	Dynamically attends to important items in the history using self-attention, with a strong focus on recent behaviors.
BERT4Rec [27]	Transformer	Leverages bidirectional learning based on masked language modeling to effectively capture long-range dependencies.
GRU4Rec [28]	RNN	Lightweight and fast, a widely adopted session-based recurrent model.
FPMC [29]	MF + Markov Chain	An early personalized recommendation model combining matrix factorization with Markov chains.
NextItNet [30]	CNN	Efficiently captures long-range patterns in sequences using dilated convolutions.
NARM [31]	RNN + Attention	Integrates attention into GRU to model both short-term and long-term user intents simultaneously.

Table 11
Five-fold cross-validation results for in-context prediction (event-based model).

Fold	Accuracy	Precision	Recall	F1-Score
fold1	0.8015	0.7769	0.7427	0.7552
fold2	0.7481	0.7287	0.7221	0.7249
fold3	0.8231	0.8236	0.7935	0.8035
fold4	0.7710	0.7558	0.7514	0.7534
fold5	0.7923	0.7621	0.7431	0.7509
Mean	0.7872	0.7694	0.7506	0.7576
95% CI	[0.7606, 0.8137]	[0.7379, 0.8008]	[0.7280, 0.7732]	[0.7352, 0.7800]

B. Supplemental information of experiments

B.1. Brief description of sequential recommendation models

Table 10 lists the six sequential recommendation models employed in our benchmark experiments. These models span a diverse range of architectural paradigms: Transformer-based (SASRec, BERT4Rec), RNN-based (GRU4Rec, NARM), CNN-based (NextItNet) and hybrid methods (FPMC) - allowing for a broad comparison of sequence modeling strategies. By including this variety, we aim to evaluate how different temporal modeling mechanisms (e.g., self-attention, recurrent updates, dilated convolutions, or Markov transitions) impact prediction performance across multiple behavioral targets (web domains, event titles, and applications). This diversity also helps identify which model families are best suited for different aspects of knowledge work prediction.

B.2. Details of five-fold cross-validation results

Tables 11 and 12 detail the fold-wise performance of the two models used in the in-context prediction task: the event-based and session-based classifiers, respectively. Inspecting these fold-level results, we observe that the event-based model exhibits relatively stable performance across all folds, with an accuracy ranging between 0.7481 and 0.8231, and the F1 score staying within a narrow band of 0.7249 to 0.8035. This consistency across partitions suggests that the model generalizes well and is not overly sensitive to variations in the training/test splits. In contrast, the session-based model shows a greater degree of variability. For example, fold2 yields substantially lower accuracy (0.6794) and F1 score (0.6272), whereas fold5 shows much stronger performance (Accuracy = 0.7538, F1 = 0.7252). This implies that the session-based model is more affected by the distribution of features across folds, probably due to its reliance on coarse aggregate features rather than sequential structure. The fold-level breakdown provides insight into the robustness and sensitivity of each model under different data partitions, complementing the averaged results presented in the main text.

Tables 13 and 14 present the fold-wise performance results for the KWA label prediction task using the event-based and session-based models, respectively. In the case of the event-based model, the performance remains relatively stable across folds, with F1 scores ranging from 0.5093 (fold3) to 0.5961 (fold2). This modest variation suggests that the model consistently captures key patterns in event sequences for multilabel classification, although some folds (e.g., fold3) may suffer from limited label diversity or skewed distributions. The session-based model, while achieving a slightly higher mean F1 score overall, shows much larger variability between folds. In particular, fold4 achieves an F1 score of 0.6041 with a very high recall (0.9162), whereas fold1 drops significantly to 0.4865. This discrepancy indicates that the session-based model is more sensitive to the distribution of co-occurring labels across folds. Its reliance on aggregated bag-of-words representations may lead to overfitting or undergeneralization depending on the composition of the validation set. These fold-level differences highlight the challenges of multilabel prediction under label imbalance and interlabel dependencies, and point to the need for stratified or label-aware data partitioning in future experiments.

Table 12
Five-fold cross-validation results for in-context prediction (session-based model).

Fold	Accuracy	Precision	Recall	F1-Score
fold1	0.7634	0.7253	0.7282	0.7267
fold2	0.6794	0.6498	0.6240	0.6272
fold3	0.7077	0.6880	0.6848	0.6862
fold4	0.6870	0.6641	0.6597	0.6615
fold5	0.7538	0.7196	0.7347	0.7252
Mean	0.7183	0.6894	0.6863	0.6854
95% CI	[0.6706, 0.7659]	[0.6481, 0.7306]	[0.6284, 0.7442]	[0.6325, 0.7382]

Table 13
Five-fold cross-validation results for KWA label prediction (event-based model).

Fold	Jaccard Score	Precision	Recall	F1-Score
fold1	0.4079	0.5843	0.5390	0.5335
fold2	0.4591	0.5770	0.6983	0.5961
fold3	0.3961	0.5359	0.5794	0.5093
fold4	0.4393	0.5445	0.6051	0.5485
fold5	0.4522	0.5468	0.6916	0.5798
Mean	0.4309	0.5577	0.6227	0.5534
95% CI	[0.3993, 0.4625]	[0.5221, 0.5933]	[0.5642, 0.6813]	[0.5161, 0.5906]

Table 15 summarizes the fold-wise evaluation results for the DBpedia entity relevance prediction task under three classification settings: binary (0 vs. (1,2)), binary ((0,1) vs. 2) and three-class (0 vs. 1 vs. 2). Across all settings, the fold-level breakdown reveals meaningful differences in task difficulty and model consistency.

- In the setting 0 vs. (1,2), the F1 scores are relatively stable across folds (ranging from 0.7571 to 0.8737), indicating that the model can reliably distinguish irrelevant entities from those with some relevance. The highest fold4 score suggests that this partition had particularly clean or separable training examples.
- In the setting more challenging (0,1) vs. 2, the F1 score varies more widely, from 0.7812 (fold3) to 0.9203 (fold4), which implies that identifying "representative" entities is more sensitive to the composition of the fold. Folds with fewer strongly representative entities may hinder classifier calibration.
- The three-class classification setting exhibits the largest performance fluctuation between folds, with F1 scores ranging from 0.5579 (fold3) to 0.8101 (fold4). This variability reflects the increased ambiguity in distinguishing relevant but non-representative entities (class 1) from the other two classes, especially when user annotations are subjective or unevenly distributed.

These fold-wise results emphasize the inherent difficulty of fine-grained entity relevance classification and suggest that future work may benefit from fold stratification with respect to entity-type distributions or additional regularization to reduce variability.

Table 14
Five-fold cross-validation results for KWA label prediction (session-based model).

Fold	Accuracy	Precision	Recall	F1-Score
fold1	0.3376	0.5143	0.6254	0.4865
fold2	0.4451	0.5960	0.7656	0.5898
fold3	0.4676	0.6228	0.8000	0.6123
fold4	0.4666	0.4894	0.9162	0.6041
fold5	0.3619	0.4175	0.8553	0.5132
Mean	0.4158	0.5280	0.7925	0.5612
95% CI	[0.3394, 0.4922]	[0.4250, 0.6310]	[0.6566, 0.9284]	[0.4900, 0.6324]

Table 15
Five-fold cross-validation results for the DBpedia entity relevance prediction task.

Recommendation Task	Fold	Accuracy	Precision	Recall	F1-Score
0 vs. (1,2) discrimination	fold1	0.7814	0.7548	0.7598	0.7571
	fold2	0.7966	0.7906	0.7950	0.7923
	fold3	0.8244	0.8457	0.8225	0.8210
	fold4	0.8776	0.8890	0.8687	0.8737
	fold5	0.8667	0.8582	0.8462	0.8516
	Mean	0.8293	0.8277	0.8184	0.8191
	95% CI	[0.7886, 0.8701]	[0.7885, 0.8669]	[0.7791, 0.8578]	[0.7804, 0.8577]
(0,1) vs. 2 discrimination	fold1	0.9023	0.9340	0.8636	0.8857
	fold2	0.9237	0.8952	0.9040	0.8994
	fold3	0.8473	0.7749	0.7884	0.7812
	fold4	0.9306	0.9163	0.9248	0.9203
	fold5	0.8095	0.8101	0.8082	0.8087
	Mean	0.8827	0.8661	0.8578	0.8591
	95% CI	[0.8377, 0.9277]	[0.8103, 0.9219]	[0.8051, 0.9105]	[0.8102, 0.9080]
0 vs. 1 vs. 2 discrimination	fold1	0.7256	0.7456	0.7245	0.7261
	fold2	0.7076	0.7063	0.7244	0.7100
	fold3	0.6221	0.5881	0.5693	0.5579
	fold4	0.8163	0.8051	0.8205	0.8101
	fold5	0.6762	0.5983	0.5759	0.5746
	Mean	0.7096	0.6887	0.6829	0.6757
	95% CI	[0.6395, 0.7796]	[0.6132, 0.7642]	[0.6125, 0.7533]	[0.5991, 0.7523]