Task as Context: A Sensemaking Perspective on Annotating Inter-Dependent Event Attributes with Non-Experts

Tianyi Li¹, Ping Wang², Tian Shi³, Yali Bian⁴, Andy Esakia⁵
¹ Purdue University
² Stevens Institute of Technology
³ Independent Researcher
⁴ Intel Labs
⁵ li4251@purdue.edu, pwang44@stevens.edu, researchtianshi@gmail.com, yali.bian@intel.com, esakia@protonmail.com

Abstract
This paper explores the application of sensemaking theory to support non-expert crowds in intricate data annotation tasks. We investigate the influence of procedural context and data context on the annotation quality of novice crowds, defining procedural context as completing multiple related annotation tasks on the same data point, and data context as annotating multiple data points with semantic relevance. We conducted a controlled experiment involving 140 non-expert crowd workers, who generated 1400 event annotations across various procedural and data context levels. Assessments of annotations demonstrate that high procedural context positively impacts annotation quality, although this effect diminishes with lower data context. Notably, assigning multiple related tasks to novice annotators yields comparable quality to expert annotations, without costing additional time or effort. We discuss the trade-offs associated with procedural and data contexts and draw design implications for engaging non-experts in crowdsourcing complex annotation tasks.

Introduction
Crowdsourcing has been widely used to annotate data for training and evaluating machine learning models. Approaches such as redundant annotations (Zaidan and Callison-Burch 2011), task decomposition (Kulkarni, Can, and Hartmann 2012), and human-in-the-loop active learning (Fang and Zhu 2014) have demonstrated success in improving annotation quality and reduce costs. Prior research has also developed successful strategies and solutions for effective task design (Yang et al. 2018) and crowd result selection and aggregation algorithms (Hsueh, Melville, and Sindhwani 2009a) to further enhance the crowd annotation outcomes. Domain-specific data annotation would additionally require careful consideration of worker expertise and provide training when needed (Servajean et al. 2016). Overall, crowdsourcing data annotation is effective but requires strategic task design and division of labor among crowd workers to optimize the annotation quality and efficiency.

When multiple interdependent data annotation tasks are conducted on the same data point, the annotations are usually conducted by separate groups as different problems and crowdsourced with workflows. For example, the presence or absence of one object may affect the classification of the object, but object detection or image classification are usually considered as different problems (Wang et al. 2011).

However, these approaches run the risk of sacrificing crucial contexts that are necessary for ensuring accurate data annotations. Furthermore, the breakdown of tasks into highly specialized microtasks may lead to confusion among crowd workers, as the overall procedural context might become fragmented or unclear. Consequently, to achieve effective crowdsourced data annotation, meticulous preprocessing techniques and expert intervention are crucial in ensuring the quality and reliability of the annotations.

In this paper, we aim to address these challenges and investigate how to enable novice crowds to conduct complex data annotation more independently, reducing the reliance on experts. We focus on the sensemaking challenges involved in data annotation and evaluate the impact of context on annotation performance. We propose a task design paradigm Task-as-Context, which models inter-dependent annotation tasks as a sensemaking process where related annotation tasks provide procedural context for each other. Prior research shows that additional data context can improve crowd performance in sensemaking tasks but may incur cognitive overload that diminishes the analysis quality (Li et al. 2019; Alagarai Sampath, Rajeshuni, and Indurkhya 2014). We draw inspiration from the prior work and investigate the potential of using Task-as-Context to onboard and scaffold non-experts in event annotation. Specifically, we aim to answer the research question: How does Task-as-Context influence non-experts' (1) performance on event annotation and (2) the annotation workload?

We use event extraction as a case example to assess the Task-as-Context paradigm, where four types of event attributes need to be annotated from unstructured text data (including trigger words, event type, event arguments, and argument roles, see Table 1). Compared with many single-task data annotations, it is necessary to comprehensively understand and capture the multifaceted nature of events along with their contextual relationships for events annotation. The complexity and interdependence of four annotation tasks in event extraction make it a unique and challenging task in NLP. We experimented with three levels of procedural context: assigning one annotation task per annotator (low procedural context); two annotation tasks per annotator (medium
procedural context); and all annotation tasks together (high procedural context). We considered data context as a confounding factor and conducted the above experiments with low and high data context. Annotation tasks with different sentences of the same event types are considered to have high data context, and tasks with sentences of more different event types are considered as having low data context.

We recruited 140 crowd participants from Amazon Mechanical Turk as non-expert annotators. They were tasked with annotating 20 sentences representing seven different event types from the ACE dataset (Table 2). The same sentence was annotated under different conditions, with a repetition of 10 annotators. This results in an aggregate of 1400 annotations for subsequent analysis. We measured the annotation quality by comparing crowd annotations to expert annotations provided in the ACE annotation guideline (Linguistic Data Consortium 2005). The annotation workload was assessed based on participants’ self-reported perception using the NASA Task Load Index (Hart and Staveland 1988) and the time spent on the annotation tasks.

Our results demonstrate that the impact of procedural context varies across different annotation tasks and is entangled with data context. Notably, the task of “identifying event arguments” exhibits the most pronounced positive effect. The annotation performance for this task improves significantly as the procedural context increases. Interestingly, we observe opposing results for the task of “identifying trigger words.” Higher data context boosts annotation performance in medium procedural context, while lower data context worsens performance in the same setup. Regarding workload, all event annotation tasks are perceived as mentally demanding and requiring substantial effort. Surprisingly, the perception of workload among the crowds does not significantly differ across different context levels. Furthermore, handling additional related tasks does not require extra time. In summary, these results offer valuable insights into the trade-offs between data and procedural contexts when leveraging non-experts in data annotations, and inform the application of the “Task-as-Context” paradigm to facilitate non-expert crowds in conducting event annotation with satisfactory quality and cost-effectiveness.

This work makes the following contributions: First, to our knowledge, this is the first attempt to evaluate the capability of non-expert crowd workers to create event annotations that include all four event attributes from unstructured sentences. Second, we propose and evaluate a task design paradigm, Task-as-Context, motivated by the sensemaking process, for interdependent data annotation with non-experts. Finally, we characterize the trade-offs of procedural and data contexts in data annotation tasks regarding annotation performance, workload, and cost-effectiveness.

Problem and Task Definitions

Event Extraction

Event extraction (EE) is a natural language processing (NLP) task that aims to detect and retrieve attributes of real-world events from unstructured natural language texts. An

| T1   | “born”        |
| T2   | BE-BORN      |
| T3   | Jane         |
| T4   | Person-Arg   |

Table 1: Four annotation tasks are conducted on an unstructured sentence. T1 identifies the trigger word (“born”) from the sentence. T2 classifies the event type (BE-BORN) of the trigger word. T3 identifies all arguments of the event. T4 classifies the role of each argument.

illustration of EE can be found in Table 1, where the sentence is annotated with four event attributes. EE is a critical building block for any application or domain that needs structured information extracted from a large corpus of unstructured data (Maisonneuve et al. 2020), such as intelligent question answering (Boyd-Graber and Börschinger 2020; Cao et al. 2020), knowledge graph construction (Wu et al. 2019; Bosselut, Bras, and Choi 2021), to name a few.

While EE is most often referred to as a sentence-level task, some research focuses on document-level event extraction (Huang and Peng 2021; Huang and Jia 2021). Document-level EE aims to extract events across different sentences within a document and tackles challenges such as extracting the scattering event arguments and holistic modeling of inter-dependency among the events in the document.

Event extraction can be closed-domain or open-domain. A closed-domain EE follows a predefined event structure, usually referred to as the event schema, that defines a set of event types and the corresponding event argument roles. An open-domain EE does not assume such a predefined event structure and the main task is detecting and clustering similar events in the text (Allan 2012; Ribeiro, Ferret, and Tannier 2017; Liu et al. 2008).

In this work, we evaluate the Task-as-Context paradigm with closed-domain, sentence-level EE annotation tasks. The results obtained can help inform future work in other types of EE annotation. Table 1 shows an example of event annotation on a sentence describing a person being born.

Task Definitions

We define the related concepts following the ACE guideline (Linguistic Data Consortium 2005). An event is a specific occurrence of something that happens at a certain time and a certain place involving one or more participants, which can frequently be described as a change of state. A closed-domain event extraction (EE) problem assumes a predefined event schema, which specifies the types of events to be annotated and the attributes of each event.

If a sentence contains a trigger word indicating the occurrence of some event, the sentence has an event mention. The trigger word is then classified to a pre-defined event type in the event schema. The entities that participated in the event are event arguments and are classified to pre-defined event argument roles associated with the event type.

In summary, closed-domain, sentence-level event annotation involves four interdependent annotation tasks (Xiang
and Wang 2019); T1. Trigger Word Identification, T2. Event Type Classification, T3. Event Argument Identification, T4. Argument Role Classification. (Table 1)

Related Work

Event Annotation Standards and Practices

The coverage and quality of annotated datasets are essential for the performance of many natural language processing tasks (Banko and Brill 2001) as well as fine-tuning and improving the performance of large language models (LLMs) on specific tasks. Most of the datasets for event extraction (Walker et al. 2006; Song et al. 2015; Aguilar et al. 2014) are manually annotated by professionals or experts with domain knowledge (Xiang and Wang 2019), guided by specific annotation standards (Walker et al. 2006; Zhong et al. 2018). While expert annotations are more accurate and reliable, the high expense limits the dataset size and event type coverage. This also makes widely evaluated, ground-truth datasets rare and pricey – it costs $4,000 for non-members to purchase the ACE dataset.

To address the above challenges, there is an increasing research interest in developing new and larger datasets without extensive expert intervention. Research on machine-generated event annotations has explored the extraction of event triggers with minimal supervision (Peng, Song, and Roth 2016; Chen et al. 2017; Reschke et al. 2014). Zero-shot and few-shot learning were used to extract both event triggers and arguments (Huang et al. 2018; Lai, Nguyen, and Dernoncourt 2020). Recently, significant attention has been directed towards the application of large language models (LLMs) in text annotation tasks (Alizadeh et al. 2023; He et al. 2023). However, it is still challenging to ensure the quality and accuracy of annotations produced by LLMs. Human evaluation and validation are still required, especially for domain-specific data annotations. Moreover, data annotation with LLMs requires significant computational recourse and prompt engineering efforts and therefore is expensive to use at scale.

Therefore, it requires a continuing effort to improve the scale, quality, and accessibility of event extraction datasets. In this paper, we explore the potential of engaging non-expert annotators in event annotation.

NLP Data Annotation with Crowdsourcing

Crowdsourcing has been widely used in data annotation for natural language applications (Mellebeek et al. 2010; Feizabadi and Padó 2014). Some NLP literature also referred to large-scale expert collaboration as crowdsourcing (Wang et al. 2020). In this work, we focus on paid, non-expert crowds from online marketplaces.

Crowd workers have been involved in a variety of text data annotation tasks ranging from word-sense disambiguation (Chklovski and Mihalcea 2003; Kapelner et al. 2012; Parent and Eskenazi 2010), entity extraction (Finn et al. 2010; MacLean and Heer 2013; Wang et al. 2012; Demartini, Difallah, and Cudré-Mauroux 2012), to affect or sentiment analysis (Brew, Greene, and Cunningham 2010; Hsueh, Melville, and Sindhwani 2009b) and even more open-ended content analysis (André, Bernstein, and Luther 2012; Benoit et al. 2016; Chilton et al. 2013). The involvement of crowd workers is especially successful in synthesizing and validating existing annotations. For example, (Wang et al. 2012) used crowdsourcing to identify and merge redundantly named entities. (Liu et al. 2016; Drapeau et al. 2016) designed a flexible workflow to validate annotations on event relations in sentences.

However, the amount of learning and sensemaking needed, as well as the inter-dependent, multi-step nature of event annotation, incurs additional challenges for crowdsourcing the process. The XLike project developed a tool to annotate events from documents (Košmerlj et al. 2014), but it remains unclear how distributed and transient crowds can use the tool to make event annotations. Some newly developed datasets used crowdsourcing for event annotation but did not cover all four types of event attributes, and did not provide details about the crowdsourcing procedures. For example, MAVEN (Wang et al. 2020) is a newly developed event detection dataset that used crowdsourcing to annotate trigger words and event types; RAMS (Ebner et al. 2019) also used crowdsourcing for event annotation but focused on event arguments and roles. The resulting annotations were also found to have mixed-quality (Zhang et al. 2022). This work builds on these prior efforts and investigates the capability of non-expert crowd workers to annotate unstructured sentences from scratch, and how to enhance the crowd annotation performance.

Crowdsourcing Paradigms

NLP data annotation is mostly crowdsourced using workflow-based paradigms. In this approach, researchers and experts break down complex tasks into smaller, manageable micro tasks and assign them to crowd workers. Although this method has proven to be highly efficient for homogeneous annotation tasks, it may encounter limitations when dealing with multiple interconnected tasks, as the potential for error propagation becomes a concern (Li et al. 2019). Alternatively, there are role-based crowdsourcing paradigms (Retelny, Bernstein, and Valentine 2017). Rather than relying on researchers and experts to decompose the problem, crowd workers would coordinate large-scale collaboration among themselves based on their roles and expertise. The role-based crowdsourcing paradigms demonstrated superior effectiveness on context-heavy data and tasks that are difficult to decompose into microtasks and require longer-term commitments, which are usually referred to as macrotasks (Haas et al. 2015).

Despite the successful application of role-based crowdsourcing paradigms in solving complex problems, such as software design and project management (Valentine et al. 2017), when it comes to intricate data annotation tasks, breaking down macrotasks into microtasks and developing a well-defined workflow leads to superior quality and more resilient outcomes (Cheng et al. 2015).

In this work, we revisited the boundary between micro and macro tasks and explore how novice crowds can contribute to multi-step annotation tasks such as event extraction.
The Task-as-Context Paradigm

Extracting event attributes from natural language data is fundamentally challenging due to the abundance of vague and indefinite expressions (Ludlow 1999). Retrieving, synthesizing, and mapping the event attributes from an unstructured sentence to a pre-defined schema requires domain knowledge and expertise. Human annotators need to grasp the predefined event schemata, which specify the rules for what event types and event attributes should be considered for annotation (“taggability” (Walker et al. 2006)), making sense of the sentences, and labeling the inter-dependent event attributes accordingly. By focusing on how individuals make sense of complex and ambiguous situations during data annotation, we formulate the relationship among the interdependent data annotation tasks as well as the data to be annotated with sensemaking theories (Figure 1).

Theoretical Foundation in Sensemaking

The Task-as-Context paradigm is grounded in sensemaking theories, especially Pirolli and Card’s sensemaking loop (Pirolli and Card 2005). It highlights the iterative nature of sensemaking in data annotation where information from unstructured sentences is extracted and schematized and iteratively refined. Specifically, the annotation process involves several subloops (Figure 1).

Firstly, T1 establishes a subloop for trigger word identification. Annotators read the sentences, extract trigger words, and iteratively determine the appropriate word based on its role in the sentence. Similarly, T3 establishes an event argument identification subloop that extracts relevant event arguments from the sentences. Both T1 and T3 align with the "read and extract" process described in (Pirolli and Card 2005). These two subloops are illustrated with the green (the bottom two) loop arrows in Figure 1. Further, T2 forms a subloop for event type classification, where annotators assign the most suitable event type to each trigger word. The event types can also be used to evaluate the accuracy of trigger word. T4 forms an argument role classification subloop, associating the event arguments with their respective argument roles defined in the event schema. T2 and T4 subloops correspond to the "schematize" process in Pirolli and Card’s sensemaking model. These two subloops are illustrated with the orange (the top two) loop arrows in Figure 1. Putting these four tasks in the sensemaking loop, T1 and T2 are consecutive sensemaking subloops where the output of T1 is passed to T2 as input. T3 and T4 are also such consecutive subloops (Figure 1).

The relationships between the subloops/tasks are further manifested in Dervin’s Sense-Making theory (Dervin 1998), highlighting the active and dynamic nature of the sensemaking process in data annotation. The gaps or discrepancies encountered in each annotation task would prompt annotators to seek information and engage in related data annotation activities – obtaining related context for conducting the annotation tasks. Each annotation task’s output informs the subsequent task or verifies/refutes preceding task outcomes (Figure 1). The trigger word identified in T1 provides context for T2 (arrow ①). The event type from T2 and the event schema determine all possible argument roles (arrow ②), providing context for T3 (arrow ③). The event arguments identified in T3 then provide contexts for T4 (arrow ④). Conversely, failure to classify an event argument into an argument role may suggest errors in T3 (arrow ⑤) or the preceding tasks, T1 and T2 (arrow ⑥). Similarly, if a trigger word cannot be classified into an event type, it may indicate an error in T1 (arrow ⑥).

The relationship between annotation rules and the data to be annotated can be characterized by Gary Klein’s data-frame theory (Klein et al. 2007). The event schema can be considered as the “frame” and the unstructured sentences are the “data”. Annotators need to develop mental models of the event schema and the sentences, draw connections between the two with retrospective sensemaking and situational awareness, and annotate the sentences with the most suitable labels. Since the “frame” (event schema) cannot be adjusted in closed-domain event annotation, we use the data-frame theory to model event annotation as an iterative process of optimizing the fitting of the data to the most suitable frame. This can be generalized to open-domain event annotation, where the “frame” (event schema) is continuously adapted based on the data.

In summary, we consider the sensemaking process of event annotation as an iterative process but with a consecutive order. An event type cannot be classified without implicitly or explicitly identifying the trigger words. Thus we consider trigger word identification as the first step in the sensemaking process, T1. And event type classification is T2. In closed-domain EE, each event type has a unique set of event argument roles. Thus, the possible event argument roles are determined once the event type is identified. However, a sentence may not contain all argument roles defined for an event type. Thus, event arguments need to be first identified and then classified to their event argument roles. Thus, event argument extraction is the third step (T3) and event argument classification is the fourth step (T4). This order is also supported by (Xiang and Wang 2019).

Levels of Task-as-Context in Event Annotation

We experiment with combining and assigning annotation tasks from consecutive sensemaking subloops to the same crowd worker and investigate the potential of these associated tasks to provide procedural context and help novice annotators to better understand the individual annotation tasks and benefit their annotation performance.
Expanding the sensemaking scope of each annotator from individual tasks to multiple tasks could potentially enhance self-agency where the annotator will continue to use the output s/he developed to conduct the next task. We hypothesize that working on more tasks in an expanded sensemaking subloop provides additional procedural context that deepens the annotator’s understanding of individual tasks and will enhance the overall annotation performance. Nonetheless, working on multiple different tasks also introduces additional challenges and distractions, which can be overwhelming and incur cognitive overload, and may undermine the potential benefits of the procedural context.

Therefore, we define three levels of procedural context for event annotation to explore how procedural context may influence the annotation performance.

- \( L_{\text{low}} \): Individual annotation tasks are conducted separately by different annotators: T1, T2, T3, T4.
- \( L_{\text{medium}} \): Two annotation tasks within the same sub-loop are conducted together: T1+T2, T3+T4.
- \( L_{\text{high}} \): All four annotation tasks are conducted together by each annotator: T1+T2+T3+T4.

## Study Design

### Experiment Variables

Our study design controls one independent variable, procedural context, and a confounding factor, data context.

The three levels of procedural context were described in the previous section. To ensure that the crowd workers only work on the annotation tasks in each procedural context level, we provided the correct answers to the related annotation tasks in \( L_{\text{low}} \) and \( L_{\text{medium}} \). For example, T2 in \( L_{\text{low}} \) did not need to identify the trigger words but instead, the correct trigger words were already identified. This unavoidably provided extra advantages to the conditions of lower procedural context levels and may overestimate the corresponding crowd performance. We take this limitation into consideration in the result analysis and also discuss future work needed in the limitation section.

We manipulate the data context by controlling the variety of event types present in the sentences of each HIT. When a HIT includes a higher number of sentences belonging to the same event types, it is considered as having a high data context. Conversely, if the sentences in a HIT encompass different event types, the data context is considered low. To control different levels of data context, we conducted two rounds of data collection. The first round has high data context, where each HIT has five sentences per event type and thus, contains two different event types. The second round has low data context with two sentences per event type and thus, contains five event types. As different numbers of event types are needed, the sentences used in the two rounds of data collection are different. We will elaborate more on this limitation in the discussion section.

There are multiple other factors influencing the annotation performance, such as the task procedure, individual differences among crowd workers, and the overall event schema. To mitigate the influence of these factors, all HITs have the same instructions and examples, the same number of sentences (\( N_s = 10 \)) to annotate, and assigned to the same number of crowd workers (\( N_c = 10 \)).

We focus on the following dependent variables: performance scores measured by precision\(^2\), recall\(^3\), and \( F_1 \)\(^4\), workload perception measured by the NASA Task Load Index (Hart and Staveland 1988), and HIT elapsed time. We use the expert-generated annotations as the ground truth to assess the annotation performance. HIT elapsed time is available from the MTurk platform.

### Dataset

Our study design implemented two levels of data context through two rounds of data collection, with the first round being a high data context (five distinct sentences per event type) and the second round employing a low data context condition (two different sentences per event type). To ensure consistency, we maintained a controlled factor of 10 sentences for each crowd worker to annotate. As a result, two different event types were needed in the first round and five in the second round. Overall, a dataset of 20 sentences and seven event types were required for the study.

We carefully selected the sentences from the ACE guidebook (Linguistic Data Consortium 2005), which contains detailed definitions and rules for each event type, and expert annotations. We based our selection on the frequency of appearance and focused on seven event types with varied coverage in the ACE dataset (Table 2). Specifically, ATTACK and TRANSPORT are among the most frequent event types covered in ACE. While MEET, START-POSITION, and TRANSFER-MONEY are with less related instances. We also selected two event types, BE-BORN and START-ORG, with the lowest frequency in ACE. These sentences were further verified by two NLP (Natural Language Processing) experts, who confirmed the sentences’ representativeness in both the ACE dataset and other event annotation datasets.

### Participants and Procedure

We hire crowd participants from Amazon Mechanical Turk (MTurk) to serve as non-expert annotators. MTurk is one of the primary paid crowdsourcing marketplaces for collecting human annotations with paid “microtasks”. On MTurk, the requester posts microtasks as Human Intelligence Tasks (HITs) and specify the number of assignments, which decides the number of repetitions of each HIT.

We configured our HITs to have 10 assignments and be visible only to registered MTurk workers who have completed more than 100 HITs with above 95% acceptance rate and have not completed any HITs from our study before. This is to avoid within-subject learning effects across different HITs, as this will provide additional procedural context. To keep each crowd task “micro”, each crowd task contains 10 sentences to annotate. Each crowd task contains four phases: 1) introduction and examples, 2) training tasks, 3) annotations, 4) final review.

\[ \text{Precision} = \frac{TP}{TP+FP}, \text{ where TP and FP denote true and false positive, respectively.} \]
\[ \text{Recall} = \frac{TP}{TP+FN}, \text{ where FN represents false negative.} \]
\[ F_1 = \frac{2\times(\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}} \]
<table>
<thead>
<tr>
<th>Event Type</th>
<th># Instances</th>
<th>Event Definition</th>
<th>Event Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>attack</td>
<td>1543</td>
<td>a violent physical act causing harm or damage</td>
<td>CONFLICT</td>
</tr>
<tr>
<td>be-born</td>
<td>&lt;100</td>
<td>a person entity is given birth to</td>
<td>LIFE</td>
</tr>
<tr>
<td>meet</td>
<td>280</td>
<td>two or more entities come together at a single location and interact with one another face-to-face</td>
<td>CONTACT</td>
</tr>
<tr>
<td>start-org</td>
<td>&lt;100</td>
<td>a new organization entity is created</td>
<td>BUSINESS</td>
</tr>
<tr>
<td>start-position</td>
<td>118</td>
<td>a person entity begins working for (or changes offices within) an organization or geographical/social/political entities</td>
<td>PERSONNEL</td>
</tr>
<tr>
<td>transfer-money</td>
<td>198</td>
<td>the giving, receiving, borrowing, or lending money when it is not in the context of purchasing something</td>
<td>TRANSACTION</td>
</tr>
<tr>
<td>transport</td>
<td>721</td>
<td>an artifact (weapon or vehicle) or a person is moved from one place (geographical/social/political entities, facility, location) to another</td>
<td>MOVEMENT</td>
</tr>
</tbody>
</table>

Table 2: Coverage of the event types from the ACE dataset. The first two event types are covered in the first round of data collection (high data context), and the remaining five are covered in the second round (low data context).

3) annotation tasks, and 4) post-task questionnaires. Each phase is rendered on a separate page. The crowd workers can withdraw by returning the HIT at any time. We estimate the time needed to complete each crowd task with a pilot study. Based on our local minimum wage, each HIT pays $2.50. We also incentivize crowd workers with a $1 bonus when more than 8 of 10 sentences are annotated correctly.

Results
A total of 140 crowd workers were recruited in the two rounds of data collection from MTurk in June and September 2022. Most crowd workers were between ages 25-65 (N=126, 90%), six participants were between ages 18-24, three were 65 or above, and five participants preferred not to reveal their ages. 72 were male (51.4%), 63 were female (45%), and five preferred not to tell (3.6%).

As the annotation tasks contained different sets of sentences, we analyzed the results of different data context conditions separately (Figure 2 and Table 3 shows results of high data context conditions; Figure 3 and Table 4 shows results of low data context conditions).

For clarity and ease of reference, we hereby refer to the crowd workers as annotators and the two rounds of data collection as Study 1 and Study 2.

RQ1: Impact on Task Performance
To assess the impact of procedural context on annotation performance (RQ1), we analyze the annotation performance with a mixed-effect model (Bates et al. 2014), where the procedural context level is the main effect. Event type serves as the blocking factor to examine if sentences from different event types had significantly different results. We consider the different sentences as the random effect to examine if the results are generalizable to a larger sentence population.

Overall, procedural context levels (main effect) significantly influenced the performance of the annotation tasks (see Table 3 and Table 4). The influences are different with different annotations tasks (T1 to T4) and different amounts of data context. Below we detail the performance of the four annotation tasks separately.

Performance of T1. Identify Trigger Word
Overall, the performance of T1 is significantly different with different procedural context levels, no matter if the data context is low or high (p < 0.05 in Table 3 and 4). Note that in T1, since each sentence contains only one trigger word, the precision, recall, and F1 values are identical.

When data context is high (Table 3), the performance of T1 shows a significant improvement in $L_{\text{medium}}$ compared to $L_{\text{low}}$, 0.05 ≤ $\beta_1$ ≤ 0.31. As the level of procedural context continues to increase, the performance continues to improve, although the effect size becomes smaller ($\beta_2 = 0.09$) and no longer statistically significant (95% CI includes 0 and negative values, indicating the possibility of $L_{\text{high}}$ performance being worse than $L_{\text{medium}}$).

Interestingly, when data context is low (Table 4), working on both T1 and T2 together ($L_{\text{medium}}$) leads to significantly worse performance than $L_{\text{low}}$ ($\beta_1 = -0.26$, $95\% CI = [-0.38, -0.14]$). Similarly, high procedural context ($L_{\text{medium}}$ and $L_{\text{high}}$) results in worse performance compared to $L_{\text{low}}$, with a smaller effect size ($\beta_2 = -0.09$) and no statistical significance ($95\% CI = [-0.21, 0.03]$).

This points to the potential interaction effect between the procedural context and data context. Having additional procedural context seems to have a positive influence on the performance of T1 only when there is high data context in the annotation task. If the data context is low, having more procedural context may increase the cognitive load of the annotator and affect the annotation performance.

Performance of T2. Classify Event Type
Similar to T1, the precision, recall, and F1 values for T2 are identical since each sentence mentions only one event type. Overall, higher levels of procedural context yield better performance for T2, without statistical significance (p > 0.5 in Table 3 and 4).

When comparing T2 performance between “knowing the answer of” T1 ($L_{\text{low}}$) and “working on” T1 ($L_{\text{medium}}$), the performance of T2 in $L_{\text{low}}$ is slightly lower than in $L_{\text{medium}}$ ($\beta_1 > 0$). This suggests that working on T1 can be as beneficial as having prior knowledge of the answer to T1.

When comparing T2 performance between $L_{\text{low}}$ and $L_{\text{high}}$, the additional procedural context does not signifi-
Table 3: The significance of the main effect (procedural context), blocking effect (event type), and the variance of the random effect (different sentences) in the high data context. As there is only one trigger word per sentence, the precision, recall, and F1 values are the same in T1 and T2. \( \beta_0 \) represents the estimated performance scores in a \( L_{low} \); \( \beta_1 \) represents the change in performance in \( L_{medium} \) compared to a \( L_{low} \), and \( \beta_2 \) represents the change in performance in \( L_{high} \) compared to a \( L_{low} \). \( \sigma^2_S \) is the variance among different sentences, \( \sigma^2_R \) is the residual variance. P-values greater than 0.05 and confidence intervals that include zero indicate insufficient statistical significance and are shown in italic fonts.

Table 4: Linear Mixed Effect Model results in low data context.

The results indicate that in both high data context and low data context conditions, the procedural context has no significant impact on T2 performance. However, the performance in low data context conditions is generally lower than in high data context conditions (see horizontal lines in Figure 2 and Figure 3), implying that the impact of data context may outweigh that of procedural context for T2. One possibility is that in low data context conditions, the greater variety of event types increased the difficulty of T2.

Performance of T3. Identify Event Argument. The performance of T3 significantly improved with higher levels of procedural context, except for the recall values in low data context conditions.

In \( L_{low} \) and \( L_{medium} \), annotators were provided with the same knowledge of correct trigger words and event types. However, in \( L_{medium} \), they also worked on T4 as part of the procedural context. The precision and F1 scores (\( T3_p \) and \( T3_f1 \)) were significantly higher in \( L_{medium} \) in both low and high data context conditions (\( \beta_1 > 0 \) and \( 95\%CI > 0 \) in both Table 3 and 4), indicating that working on T4 as a procedural context enhanced the performance of T3.

Similarly, in \( L_{medium} \) and \( L_{high} \), annotators worked on T3 and T4 together in both conditions. In \( L_{medium} \), they were provided with the correct trigger words and event types, while in \( L_{high} \), they had to identify the trigger words and event types themselves. The precision and F1 scores were higher in \( L_{medium} \) compared to \( L_{high} \) (see T3 Precision and F1 graphs in Figure 2 and 3), suggesting that for T3, it was more beneficial to have the answers to T1 and T2 directly than to work on those tasks.

Comparing the performance of T3 in \( L_{low} \) and \( L_{high} \), the knowledge of the correct trigger words and event types (\( L_{low} \) did not benefit the precision and F1 scores (\( \beta_2 > 0 \) and \( 95\%CI > 0 \) in both Tables 3 and 4). In other words, knowing the answers to T1 and T2 \( (L_{low}) \) was not as helpful as working on T4 as part of the procedural context \( (L_{high}) \). These findings highlight the importance of T4 as a crucial procedural context for T3, contributing to its improved performance.

Performance of T4. Classify Event Argument Roles. Comparing the performance of T4 in \( L_{low} \) and \( L_{medium} \), annotators were provided with the correct answers for T1 and T2, but \( L_{low} \) also included the correct answer for T3, while \( L_{medium} \) required annotators to identify the answer for T3. The mean performance metrics were higher in \( L_{medium} \) in both low and high data context conditions (\( \beta_1 > 0 \) in both Table 3 and 4). Although only the recall in \( L_{high} \) showed sta-
RQ2: Impact on Annotation Workload

We measure the annotation task workload with two metrics: the self-reported perception using the NASA-TLX survey (Hart and Staveland 1988) and the Human Intelligence Task (HIT) elapsed time on MTurk. Unlike the performance analysis for RQ1, the perceptions and elapsed time are collected for each HIT on MTurk, and thus cannot be analyzed separately for each annotation task (T1, T2, T3, T4). Therefore, the analysis was conducted at the level of HITs, rather than sentences.

Self-Reported Workload Perceptions

We examined responses to the NASA-TLX survey (Hart and Staveland 1988) using a linear mixed-effect model and chi-squared test. Overall, the annotators perceived the event annotation tasks as neutral or low physical and temporal demand, while efforts and mental demand were rated somewhat high, regardless of the task and data context levels. Interestingly, the workload perception was not significantly influenced by the task and data context levels. The Chi-squared Test of Independence confirmed that perceptions were independent of the task conditions. In other words, the non-expert annotators did not perceive the different types or numbers of annotation tasks differently.

Time Spent on annotation tasks

The HIT elapsed time represents the duration between HIT acceptance and submission on MTurk. It’s important to note that this elapsed time may include breaks if annotators left the task page open. Additionally, Phase 1 introduction and examples and Phase 4 post-task questionnaires are identical across all annotation tasks, and Phase 2 training tasks and Phase 3 annotation tasks vary based on the corresponding annotation tasks. Therefore, we compare the differences in HIT elapsed time rather than absolute values.

The HIT elapsed time (in minutes) did not show significant differences among the different conditions ($p > 0.05$). The overall time required to complete all four tasks ($L_{high}$) did not exceed the maximum time needed for individual tasks ($L_{low}$).

In high data context conditions, working on T1 and T2 together (mean elapsed time = 21.34 minutes) took less time compared to the sum of working on each task individually (mean elapsed time for T1 = 12.54 minutes, mean elapsed time for T2 = 11.01 minutes). Similarly, in low data context conditions, working on T1 and T2 together (mean elapsed time = 20.02 minutes) required less time than working on T2 alone (mean elapsed time = 25.97 minutes).

The time spent working on T3 and T4 together was only slightly longer than working on T4 alone. Considering these findings along with the fact that T3 had the highest HIT elapsed time, we conclude that T3 is more challenging for non-expert annotators, but working on T4 helped them understand and work on T3 more effectively.

Discussion and Future Work

In this work, we propose the Task-as-Context paradigm, viewing event annotation as a sensemaking process. We explore the effectiveness of inter-dependent annotation tasks to provide procedural contexts to guide non-expert annotators through event annotation. The study results demonstrate a positive impact of procedural context on enhancing the over-
all annotation performance of non-experts, without imposing additional task time or increasing the perceived workload. Our findings also shed light on the trade-offs between procedural and data contexts in complex data annotation.

**Sensemaking Context for Data Annotation**

Our findings highlight the positive impact of assigning multiple annotation tasks to the same annotator. As the relevance between tasks decreases within the sensemaking process, the degree of improvement may diminish accordingly. For example, working on T3 and T4 together led to a substantial increase in T3 performance ($L_{\text{medium}}$ vs. $L_{\text{low}}$). If we continue increasing the procedural context by also adding T1 and T2, which are less relevant to T3 in the sensemaking process (Figure 1), the additional procedural context did not result in a significant increase in T3 performance.

Another interesting observation is that the impact of T4 on T3 is more substantial than the impact of T3 on T4, indicating that the impact of procedural context can be directional. This might be related to the role of the tasks in the sensemaking process (Figure 1). From the lens of the sense-making loop theory (Pirolli and Card 2005), T1 and T3 are lower-level information foraging tasks while T2 and T4 are schematizing tasks that classify the outputs of T1 and T3 into a predefined category. For example, having the goal of classifying the event type (T2) in mind can help with the identification of the trigger words (T1), and having the goal of classifying the event argument role (T4) in mind can help with the identification of the event arguments (T3). As the perception ratings and HIT elapsed time are not significantly different, the effort spent on one task alone is similar to that on two related tasks together. We reckon that the same amount of effort was better allocated to working on a related annotation task that helps understand both annotation tasks. In summary, information foraging tasks are more sensitive to procedural context and benefit from guidance from neighboring schematizing tasks.

**Types and Granularity of Contexts**

The Task-as-Context paradigm explores the boundaries of micro and macro tasks and suggests the tasks themselves as another type of context. By examining task decomposi-
tion at a procedural level, Task-as-Context sheds light on the value of recombining some related tasks together. Motivated by the sensemaking theories, Task-as-Context combines microtasks within the same sensemaking sub-loops (Figure 1), where the output of one task is used as the input of another task, as procedural context for non-expert annotators. Our findings indicate that the inclusion of additional tasks within the same sub-loops can effectively support non-expert annotators in producing high-quality annotations and generating a greater number of outcomes. Importantly, this approach avoids the need for annotators to engage in extra context switching or mentally transition to another sub-loop.

Task-as-Context differs from the role-based structures for macrotasks in that it does not require domain expertise or long-term commitment. It supports non-expert annotators without domain expertise to make meaningful contributions with a small amount of time and effort. Yet, unlike typical microtasks, Task-as-Context engages crowd workers in a more comprehensive sensemaking experience. The paradigm can be used with other microtask workflows to solve complex problems. For example, it can expand the scope of each task in an iterative workflow (Chilton et al. 2013) or merge some neighboring tasks in a crowdsourcing pipeline (Li, Luther, and North 2018).

Our findings also contribute to a deeper understanding of the demarcation between macro and microtasks. The results of T1. Identify Trigger Words in high data context and low data context suggests that the amount of data context may confound with the impact of procedural context. For simpler problems, microtasks with less procedural context can produce higher-quality outcomes and a better experience (Cheng et al. 2015). For more expertise-demanding tasks, the advantages of procedural context become more evident. Using tasks as context could also save the time and cost needed in the crowdsourcing process without hurting the overall performance.

Implications for Applying Task-as-Context

While the Task-as-Context paradigm is proposed for event annotation with novice crowds, we see opportunities for applying this paradigm in other multi-step, interdependent crowdsourcing problems. Below we draw implications for using Task-as-Context in other crowdsourcing contexts.

Scoping and selecting procedural context. When crowdsourcing a complex problem with multiple constituent tasks, researchers can consider combining consecutive tasks in one crowd task. The task adjacency or distance needs to be determined by domain-specific process models. Tasks that focus on information foraging are especially suitable for applying the Task-as-Context paradigm.

Formatting procedural context. We provided procedural context by asking crowd workers to work on the related tasks. Procedural context can also be delivered through other formats, such as reviewing, validating, or critiquing the work by other crowd workers. Future research can compare the efficacy of different ways of providing procedural context.

Balancing data and procedural context. Relevant tasks can serve as effective context to onboard and scaffold novice crowds in complex tasks but might suffer from diminishing returns. Increasing procedural context might unavoidably decrease data context or increase workload. Increasing the overall workload may overwhelm the crowd workers and overshadow the benefit of the additional context. When the tasks are relatively intuitive, and the main goal is to “connect the dots”, researchers might want to prioritize more data context. When the tasks require some learning and are less familiar to the crowd workers, providing additional procedural context might enhance the performance more effectively.

Limitations and Future Work

To ensure that all crowd tasks contain the same number of sentences, we had to control the confounding factor with two rounds of data collection. The different event types, sentences used, and date and time of data collection, can all introduce extra variance to the results. To address this limitation, we analyzed the results with mixed-effect model and considered individual sentences as the random factor, to examine the generalizability of the results to other sentences.

Despite the experimental dataset containing representative sentences and being vetted by NLP experts, its size remained relatively small. This decision was driven by our goal to compare various annotation strategies, which requires amassing a substantial number of annotations for identical sentences under diverse conditions. We successfully collected a total of 1400 annotations to answer our research questions. Future investigations could explore deeper into confounding factors, such as the diversity of sentences and event types, and/or the volume of sentences per HIT.

To ensure annotators only work on the annotation tasks in each procedural context level, we provided the correct answers to the related annotation tasks in $L_{low}$ and $L_{medium}$. This provided extra advantages to the conditions of lower procedural context levels and might have overestimated the corresponding annotation performance. While error propagation can happen both between different annotators or within the same annotators across different tasks, future research is needed to empirically examine the error propagation within high procedural context HITs vs. across low procedural context HITs.

Broader Perspectives

Improving the engagement of non-expert annotators can have far-reaching benefits, not only in terms of expanding the scale and efficiency of data annotation but also in enhancing the overall quality and reusability of datasets. Non-expert annotations can play a crucial role in assessing existing datasets by identifying potential errors, biases, or inconsistencies. Additionally, supplementary soft labels can be derived from the collective input of multiple non-expert annotators, complementing the existing hard labels.

Crucially, reducing barriers for non-experts to contribute to the data annotation process encourages a diverse range of perspectives to shape the training data for AI systems. This helps mitigate the risks associated with reinforcing biases that may arise from relying solely on a minority of experts or annotators.


