Automating Annotation Guideline Improvements using LLMs: A Case Study

Adrien Bibal[1], Nathaniel Gerlek[1], Goran Muric[1], Elizabeth Boschee[2,*], Steven Fincke[2,*], Mike Ross[3,*], Steven N. Minton[1]

[1]InferLink Corporation

[2]Information Sciences Institute (ISI), University of Southern California

[3]Meta AI

[*]These authors equally contributed to this work and are listed by alphabetical order.

{abibal,ngerlek,gmuric,sminton}@inferlink.com,

{boschee,sfincke}@isi.edu, mikeross@meta.com

Abstract

Annotating texts can be a tedious task, especially when texts are noisy. At the root of the issue, guidelines are not always optimized enough to be able to perform the required annotation task. In difficult cases, complex workflows are designed to be able to reach the best possible guidelines. However, crowdsource workers are commonly recruited to go through these complex workflows, limiting the number of iterations over the workflows, and therefore, the possible results because of the slow speed and the high cost of workers. In this paper, our case study, based on the entity recognition problem, suggests that LLMs can help produce guidelines of high quality (inter-annotator agreement going from 0.593 to 0.84 when improving WNUT-17's guidelines), while being faster and cheaper than crowdsource workers.

1 Introduction

Designing guidelines making every annotator agree on their annotations is a difficult and tedious process. Such a task can be even more difficult in the context of noisy texts, common in fast-paced online communication especially on platforms such as X (formerly known as Twitter). Such texts can be filled with typos, with specific tokens (e.g., Twitter handles and hashtags in the context of Tweets) and with interjections (such as, for instance, "mmmmhhhh"). Furthermore, texts to annotate can lack context, because of the nature of the text (e.g., an isolated Tweet) or because of its collection (e.g., when the logical connection between the elements of a discussion has been lost).

These challenges can lead to many iterations over the guidelines to clarify to the annotators how to annotate. The classic iterative workflow is, (1) to have an expert designing the guidelines, (2) to have annotators annotating with these guidelines, (3) to compute an inter-annotator agreement (IAA) to check if the guidelines are clear enough. If there are not clear enough, Steps 1 to 3 are performed again until an acceptable IAA is reached (Pustejovsky and Stubbs, 2012).

Expense and time issues related to the multiple iterations over the workflow are commonly reduced by working with crowd workers (when compared to, e.g., in-premise recruitment). However, this is sometimes not enough when working with complex workflows (Pradhan et al., 2022). Indeed, the number of iterations to reach good guidelines can be very large, and processes sometimes never converge.

In this paper, we suggest via a case study that complex annotation workflows can be automated with large language models (LLMs), producing quality guidelines, while significantly reducing the cost and time needed to go through such workflows (more than 700 times cheaper and more than 300 times faster). Furthermore, using LLMs to automate such workflows also has the benefit of avoiding human biases, such as post-rationalizing to stick to their choices when suggested to reconsider them.

2 Related Work on the Optimization of Guidelines

This section introduces the related work on optimizing guidelines using complex annotation workflows. Please note that we use, as done in the literature, the term "workers" to refer to the people involved in the workflow used to optimize the guidelines. Indeed, "annotators" only corresponds to the subset of workers who perform the annotation work in the workflow. Also, note that we do not restrict our presentation of the literature to annotations in natural language processing, as relevant workflows have been proposed to annotate other items than sentences or words, such as

Proceedings of Context and Meaning: Navigating Disagreements in NLP Annotation, pages 129–144
January 19, 2025.

images and websites.

When dealing with the task of improving annotation guidelines, the classic approach is MAMA, or Model-Annotate-Model-Annotate (Pustejovsky and Stubbs, 2012). The idea is simple: the annotation task needs to be modeled, through guidelines, then evaluated via a proper annotation task, from which will follow a revision of the guidelines, and so on. In order to evaluate the guidelines, the agreement between the annotators (or IAA for inter-annotator agreement) is generally used (Pustejovsky and Stubbs, 2012).

Other, more complex, workflows have been developed over time, but with an ever increasing investment in time and money. Bernstein et al. (2010) proposed a Find-Fix-Verify workflow. This workflow can be adapted to very different scenarios, but in our context, Find (the first step) corresponds to asking workers to find ambiguous elements in the guidelines given some examples to annotate. Based on the identified issues in the guidelines, workers, in the Fix step, propose alternatives to each problematic element in the guidelines. The last step, Verify, then consists in asking new workers to vote for the best alternative.

Drapeau et al. (2016) later introduced a Justify-Reconsider workflow that leverages rationales from the workers. In the Justify phase, workers provide rationales for their annotations. Then, after reading the rationales from other workers, each worker is given the possibility to reconsider their annotations. While this workflow can provide more accurate annotations, it stops short of improving the guidelines.

In a similar fashion, Chang et al. (2017) proposed a Vote-Explain-Categorize workflow that also leverages rationales. The first stage, Vote, is the annotation stage, with the addition of an option for the annotators to express their uncertainty. The examples showing disagreement or uncertainty are then selected for the Explain step, where workers are asked to provide a rationale for these selected labels. Finally, the Categorize stage consists, for each worker, in choosing a label based on the explanations.

In both their work, Drapeau et al. (2016) and Chang et al. (2017) noted the difficulty of obtaining quality rationales in the workflow. Wang et al. (2018) developed a solution that they called "Rewarding the Brave" to pay workers based on the effectiveness of their rationales in convincing other workers.

Instead of only asking for a rationale, a discussion between the workers using a chat platform can also be envisioned (Schaekermann et al., 2018; Chen et al., 2019). This solution has been found to be effective, but comes with a significant increase in time.

Bragg et al. (2018) proposed a Work-Filter-Diagnose-Clarify/GenTest-Organize-Refine workflow. The first stage, Work, corresponds to the annotation step in our case. Based on this annotation work, examples causing disagreement are selected in the Filter stage. Then, in the Diagnose stage, another set of workers analyze the disagreement on each example to identify if it is best to clarify the guidelines (Clarify stage) or to add the problematic examples to the guidelines (GenTest stage). In the Organize stage, a clustering approach then automatically organizes the various propositions made by the new set of workers to clarify the task. Finally, in the Refine stage, the guideline makers then take inspiration from the worker's propositions to improve the task and the guidelines.

WingIt is a solution to spot ambiguous cases and to propose improvements to the guidelines (VK Chaithanya and Quinn, 2018). The workers have the possibility to ask questions and to propose answers to these questions. The guideline makers can then choose to pick from the suggested answers, or make their own.

In a subsequent work, VK Chaithanya et al. (2019) proposed TaskMate, which is a 5-stage workflow: Identify-Resolve-Merge-Verify-Select. The Identify stage corresponds to the questions and answers of WingIt. However, instead of the guideline makers having to evaluate and select an answer, the workers themselves vote, in the Resolve stage, for the best answer to each question. Based on all the votes, the workers are then asked to propose new guidelines in the Merge stage. In the Verify stage, the workers have to check if the new proposed instructions indeed clarify the original ambiguities. Finally, among all the newly proposed instructions that pass the check, the workers have to vote again, in the Select stage, for the improved instructions that will be included in the new version of the guidelines.

Finally, directly inspired by the Find-Fix-Verify workflow of Bernstein et al. (2010), Pradhan et al. (2022) proposed a Find-Resolve-Label workflow. The Find stage is similar to the one of Bernstein et al. (2010). In the Resolve stage, the guideline

makers select some of the ambiguous examples and integrate them as examples in the guidelines. The Verify stage is then an annotation task with the guidelines and the ambiguous examples.

Note that focusing on improving the guidelines may not be the only solution to the problem. Chen and Zhang (2023) showed that two dimensions can be considered when dealing with the problem: (1) how much the texts to annotate are ambiguous and (2) how much the guidelines are ambiguous. If the texts to annotate are ambiguous, the solution can be to modify the texts themselves. However, if text ambiguity is not the main issue, then the guidelines probably are. It may then be worth improving the guidelines. Note that the solution proposed here can only be applied if modifying the texts to annotate is an option.

Many, if not all of these workflows, require multiple iterations, which is not doable in practice. While cost and time are regular concerns for these workflows, the new advances with large language models (LLMs) may allow for a solution: swapping crowdsource workers with LLMs. However, the question is: can LLMs go through a complex workflow and produce good guidelines? We propose in this paper a case study exploring this question. However, before that, we present in the next section the workflow developed for our case study.

3 Workflow Used in our Case Study

While some parts of the literature showed that developing and using complex workflows with workers are costly and time-consuming, other parts highlighted the advance of LLMs in the domain. Gilardi et al. (2023) showed, for instance, that GPT models outperform crowdsource workers in terms of accuracy on annotation tasks. At the same time, it has been shown that LLMs can follow annotation guidelines closely, and their agreement is on par with the agreement of human annotators among themselves (Fonseca and Cohen, 2023). Furthermore, in the case where pre-trained LLMs are not good enough at following guidelines, Sainz et al. (2024) have shown that LLMs can be fine-tuned to be specifically better at that.

This section introduces the workflow developed for our case study. We present in Section 3.1 the different phases of the workflow. Section 3.2 will then discuss its automation.

3.1 Annotate-Justify-Reconsider-Fix

Inspired by the literature and various internal tests, we developed Annotate-Justify-Reconsider-Fix as a pattern for the workflow in our case study (see the left part of Figure 1). The first phase of the workflow, "Annotate", is self-explanatory: annotators are first asked to annotate, given certain guidelines.

In the "Justify" phase, each annotator is asked to justify their annotations (or lack thereof) in two situations: (1) if there is a disagreement on an annotation, and (2) if the annotator did not annotate an element, while other annotators did. This phase is separated from the first one for two reasons: (1) because asking for a rationale is not necessary if everyone agrees on the annotation, and (2) because what is not annotated, but could be, is only known after some annotations are provided. During this phase, the workers can change their mind about their annotation.

In the third phase, "Reconsider", the annotators are asked to reconsider their annotations considering the other annotators' rationales. In addition to choosing their final annotations, the annotators are asked to suggest changes to the guidelines. Seeing different arguments for the same annotation often makes ambiguities in the guidelines more visible.

Finally, in the "Fix" phase, the annotators are asked to compile their suggested changes to the guidelines to re-write the guidelines. The objective of this phase is twofold. First, the guideline makers do not have to interpret the annotator's suggestions to perform the changes. Second, suggestions are given, during Phase 3, per annotation. This means that annotators did not necessarily have the big picture in mind when they provided their suggestions. Because of that, some suggestions may be contradictory. The annotator is therefore the best person to provide the global changes that best reflect the sum of their local changes.

After the merge of all the guidelines proposed in the Fix Phase by the workers, a new iteration over the workflow can begin. This process continues until a desirable inter-annotator agreement (IAA) is reached.

Now that the different components of the workflow have been presented, next section will describe how they are automated with LLMs.

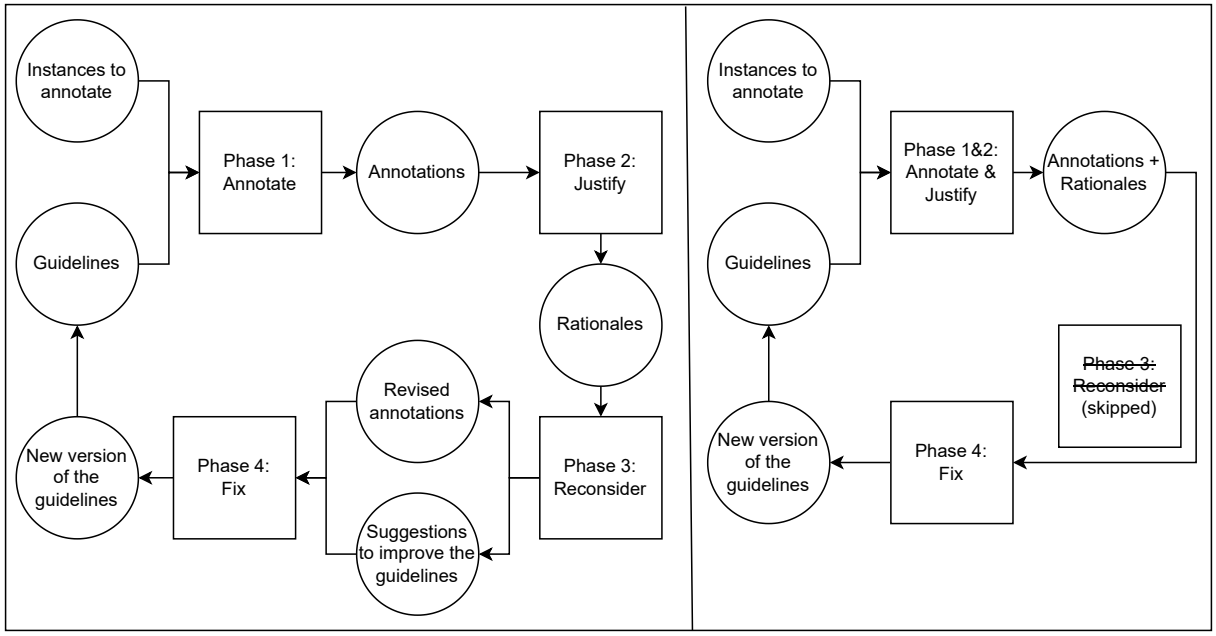

Figure 1: On the left, the workflow developed for our case study with workers. On the right, version of the workflow adapted for automation.

3.2 From Workers to LLMs

In order to automate the workflow, LLMs are used replace workers in all phases of the workflow, as well as the staff member optimizing the guidelines. To do so, two types of LLMs are used: annotator LLMs and an optimizer LLM. We define the annotator LLMs as models that are developed to provide annotations based on particular guidelines and an input text to annotate. On the other hand, we define the optimizer LLM as the model doing the task of the guideline makers: collecting guideline suggestions in order to define new guidelines for the next iteration of the workflow.

One key element to mention in the automation is that LLMs do not have the same attention issues and biases as people. For instance, we noted in our internal tests a clear bias from people to stick to their previous choices. Indeed, some annotators seem to prefer to post-rationalize their choices, even when difficult to defend, rather than to admit that they may have made a mistake. While LLMs have their own attention issues and biases (e.g., forgetting long-distance context because of their architecture and/or training), these are definitely not the same as the ones of people (e.g., being inattentive due to their fatigue). This has implications, mainly, on the "Justify" and "Reconsider" phases. During preliminary internal tests, we could see that an important advantage of Justify is that it forces workers to double-check their annotations. However, LLMs generally do not need this double-check.

Another difference between the workflow for people and for LLMs is that Phases 1 and 2 are mixed for LLMs. This is because the time required to write a rationale for each annotation is far less of an issue than it is with people. Also, asking LLMs for a rationale actually improves their annotations, as it is similar to a chain-of-thought strategy (Wei et al., 2022).

Concerning Phase 3, "Reconsider", we observed that people can have very different levels of expertise and knowledge when annotating. "Reconsider" is therefore quite important when an "expert" among the annotators can provide a rationale that will convince the other annotators. While the variance in expertise on different subjects can be high between human annotators, current pretrained LLMs have a high degree of knowledge across the board. We noticed during our internal tests that the Reconsider phase was not that important for LLMs, while it was quite critical for human annotators. Because of that, we removed the LLM calls to reconsider in the automated workflow. The LLM-based workflow is shown in the right part of Figure 1.

Based on this workflow and its automation, we present an experiment in the next section. The goal of our case study is to discuss the quality of guidelines that can be produced by such a methodology,

while also giving some numbers on the increase in speed and decrease in cost when using LLMs.

4 Experiment

This section presents an experiment, for our case study, comparing workers and LLMs going through the workflow. In order to make this comparison, we first explain our experimental setup in Section 4.1. We then go through the details of the annotation work done with workers on a Prolific, crowdsourcing platform, in Section 4.2. Finally, we present our insights in Section 4.3.

4.1 Experimental Setup

In order to give a fair shot to the LLM considered in this case study, we use one of the best models to this date[1]: GPT-4o-08-06 [2]. The temperatures of the annotator LLMs and the optimizer were set to 0.9 and 0 respectively. The prompts are provided in Appendix A.

To compare the automation of our workflow with workers going through it, we consider the entity recognition (ER) problem. In ER, the goal is to detect entities in sentences in order to then assign one or more categories to them.

As the sources of the problem can both be in ambiguities of the texts to annotate and in ambiguities of the guidelines (Chen and Zhang, 2023), we consider a dataset that expresses these two issues: WNUT-17 (Derczynski et al., 2017). The dataset features ambiguous sentences such as Tweets, along with guidelines that are not entirely optimized to annotate these ambiguous sentences. 20 sentences selected at random, and containing multiple entities, were considered for the experiment.

Next section presents how workers have been recruited to go through our workflow with the ER problem applied to the WNUT-17 dataset.

4.2 Crowdsourcing

The workflow developed for this case study has requirements to consider when choosing a crowdsourcing platform. First, the same workers are sometimes required to work on connected phases. For instance, the "Justify" phase requires that the workers justify their annotations provided during the "Annotation" phase. Second, as the workflow is quite complex, a platform that allows the staff member managing the annotation (or annotation manager) to redirect workers to some forms, spreadsheets, etc. outside the platform is necessary. Based on these constraints, Prolific [3] was chosen. The only filter used to recruit workers was their fluency in English. A bonus payment was provided to incentivize workers to do a meaningful job (as suggested by "Rewarding the Brave" of Wang et al. (2018)).

4.3 Insights

This section presents the insights gained from the comparison between the worker-based and LLM-based workflow. We perform our case study in two parts: correctness insights, and time and cost-related insights.

In order to get the data needed to perform this comparison, workers went through the workflow once, and then redid the Annotation phase. 16, 10, 8 and 12 workers went through, respectively, Phases 1, 2, 3 and then Phase 1 again.

Note that the Fix Phase (Phase 4) had to be simulated by us, because only 2 workers provided one suggestion each to change the guidelines. We hypothesize that this issue is due to the low motivation of workers on crowdsourcing platforms (despite the possibility of receiving a bonus). We elaborate on this in Section 5.2. Our simulation of what the workers could propose as changes to the guidelines revolved around the question "what changes the workers would have wanted to see in the guidelines when they wrote their arguments?"

4.3.1 Correctness Analysis

In order to get some insights about the correctness of the LLM-produced guidelines, we propose a qualitative and a quantitative assessment. First, we compare the two solutions after one iteration. Second, we produced a new round of annotations with both the original and the LLM-produced guidelines.

Let us start with the comparison after one iteration. The rationale behind this test is that, first, each iteration over the workflow is independent to the others, and can therefore be analyzed separately. Second, the improvement of the guidelines is front-loaded – most of the changes are performed at the beginning. Comparing the worker-based and LLM-based solutions on the first im-

[1] As per LMArena (https://lmarena.ai/?leaderboard) at the time of the experiment.

[2] https://platform.openai.com/docs/models/gpt-4o

[3] https://www.prolific.com/

proved guidelines can give us an idea of their respective amount and quality of changes. The comparison between the worker-based and LLM-based solutions is shown in Table 2 and Table 3 of Appendix B.

When comparing the guidelines, it can be seen that the LLM-based solution not only added elements to the guidelines, but also provided a lot of reformulations and additional examples. For instances, fully circular definitions like "*location: Names that are locations*" have been replaced by more meaningful descriptions, such as "*location: Names that are specific geographic locations or landmarks*". In the case of the worker-based solution, the changes are very localized. This is due to (1) workers not really providing suggestions to change the guidelines, and due to (2) the difficulty to make changes by having all examples of the dataset in mind. These points have repeatedly been shortcomings in the internal tests we made.

To obtain a quantitative understanding of the quality of the LLM solution, we conducted an additional experiment. The objective of this experiment is to compare the IAA produced by the original guidelines with the guidelines produced after the last iteration over the workflow by the LLMs. Because making sure that the guidelines are followed is paramount in this experiment, we decided to not rely on crowdsource workers. Instead, we mobilized 8 staff members, and each was instructed to annotate given two sets of guidelines (the original WNUT-17 guidelines before starting the workflow and the LLM-improved ones), while paying close attention to the guidelines. The LLM-improved guidelines can be seen in Table 4 of Appendix C.

The results of this experiment are shown in Table 1. One initial observation is that the IAA is only barely better with the LLM-based guidelines when compared to the original ones, before iterating over the workflow (first row of Table 1). This is due to three issues creating disagreements independently to the quality of the guidelines: intrinsically unclear entities, annotators' lack of knowledge about entities and inattention mistakes.

The issue related to intrinsically unclear entities is well-known in the literature (Chen and Zhang, 2023). In some situations, the context does not help annotators decide for their annotation, e.g. in "*Stairs : po jaket MU sampai tgl 8 jan IDR 175rb @Bagusr18971897 PIN 32783FC8 SMS 081912233358*". This sentence will often lead to disagreements, even when very good guidelines are considered. By analyzing all the sentences to annotate and the entities identified by the annotators, we tagged all intrinsically unclear entities. After removing such entities (25 left over 39), the IAA of the original and LLM guidelines become 0.558 and 0.613 respectively. As can be seen, the original gap in IAA of 0.011 enlarges to 0.055 when dropping this source of disagreements.

Annotators' lack of knowledge about certain entities is another important source of disagreements. Indeed, disagreements between annotators can occur when annotators lack the relevant knowledge. In order to identify the disagreements that were caused by a lack of knowledge, we interviewed the annotators based on annotations that seemed odd. We spotted these odd annotations by identifying all the entities for which an internet search could easily clarify what the entity is. For instance, "*Real*" in "*RT @KaiWayne : I think Big Sam was misquoted when he said he could manage Real. What he actually said was he could manage a real ale.*" corresponds to Real Madrid (the football club), but one annotator annotated it as a Person. The reason for this particular annotation is that, without knowing that Real is Real Madrid, one can see Real as a singer who is managed by Big Sam. Dropping the entities where at least one annotator showed a lack of knowledge (26 left over 39) leads to IAAs of 0.441 and 0.474 (gap of 0.033). Note that the IAAs are lower than when all entities were considered. This is because entities with a high agreement can be dropped because only one annotator showed a lack of knowledge.

Finally, inattention mistakes is another issue that causing disagreements. Even if the guidelines are perfect, annotators can miss entities to annotate, and can also miss or forget particular instructions in the guidelines. During the interview mentioned above, we became aware of and noted some inattention mistakes made by the annotators. When dropping the entities with attention issues (34 left over 39), the IAAs become 0.463 and 0.478 (gap of 0.015). Again, like in the case of the lack of knowledge, some entities were dropped despite having high agreement, explaining the lower IAAs.

A final overview of the true impact of optimizing the guidelines can then be provided by dropping entities belonging to any of these three issues (11 left over 39). By doing so, the disagreements

Selected Entities	Original Guidelines	LLM-based Guidelines
All entities	0.488 [0.484, 0.491]	0.499 [0.496, 0.502]
All entities, excluding intrinsically unclear entities	0.558 [0.555, 0.562]	0.613 [0.608, 0.617]
All entities, excluding lack of knowledge	0.441 [0.436, 0.446]	0.474 [0.469, 0.479]
All entities, excluding inattention mistakes	0.463 [0.46, 0.466]	0.478 [0.475, 0.482]
All entities, excluding all 3 issues	0.593 [0.588, 0.599]	**0.84 [0.836, 0.845]**

Table 1: Comparisons, in different situations, of the inter-annotator agreement (Fleiss' Kappa) between the original WNUT-17 guidelines (before iterating over the workflow) and the ones improved by the LLM-based solution after 4 iterations. 95% confidence intervals calculated by a bootstrap sampling with 1,000 samples are provided.

that are compared are mainly about the changes in the guidelines, and less about issues independent to the quality of these guidelines. In that situation, the IAAs for the original guidelines and the LLM-based guidelines are 0.593 and 0.84 respectively (gap of 0.247).

It therefore seems like LLMs can produce guideline changes that greatly reduce the disagreement between annotators. However, it also seems like the benefit of these changes can be hidden by disagreements caused by other issues. Each of these issues must therefore be handled alongside the guidelines.

4.3.2 Cost and Time-related Analysis

While it seems evident that the LLM-based solution saves money and time, when compared to the worker-based solution, we conducted experiments to quantitatively assess this gap. Indeed, while it is intuitive that LLMs are faster and cheaper, we argue that it is important to be able to put numbers on these intuitions.

During these experiments, we could observe that going through the annotation workflow with LLMs was more than 300 times faster and more than 700 times cheaper than with crowdsource workers. Many details about these experiments, including the time and cost per phase, can be found in Appendix D.

5 Discussion

While the insights provided above had the objective to shed more light on the difference between crowdsource workers and LLMs going through complex annotation workflows, this section focuses on additional points to discuss.

5.1 Annotation Manager's Time

In addition to the time and cost of the task itself, a non-negligible time is also spent by the annotation manager on tangential sub-tasks, such as coordinating the workers, ensuring that everything goes smoothly, checking their work (and acting when cheating is found), answering messages, etc. None of these sub-tasks are required when working with LLMs.

However, one can argue that the time needed to code and debug the LLM-based solution is, on the other hand, not required for the worker-based solution. A counter-argument to that would be that implementing the LLM-based solution is needed once, while coordinating/managing workers is to be done every time workers work with the workflow.

In both solutions, though, it is difficult to measure the required time. For instance, assessing the time needed to implement the LLM solution would require several coders coding the solution from scratch and taking their average time as an estimate. We leave this analysis as a future work.

5.2 Quality of Workers' Work

The poor quality of work in crowdsourcing platforms, as well as cheating, is well known and documented in the literature (Gadiraju et al., 2015; Xia, 2024). This kind of behavior, seen in multiple occasions during our study, has three main consequences in our context.

First, poor quality guidelines are obtained, which increases the number of iterations over the workflow that are needed to reach good quality guidelines and labels. Second, low effort can sometimes be hard to detect. Indeed, our study is based on the fact that texts can be noisy and an-

notating can be difficult. Because of that, it is difficult to differentiate between semi-random annotations and honest annotations misled by the noisy nature of texts. Third, a significant amount of the annotation manager's time must be spent detecting and acting on these cases (see the previous section). This has the effect of indirectly increasing the cost of the worker-based version of the workflow. We argue that these consequences can be avoided when using LLMs.

5.3 Speed and Cost Improvement of LLMs vs. Crowd Workers

While our study is a snapshot in the history of LLMs, current trends indicate that the performance of LLMs will continue to evolve and improve. Along with their quality, their speed and cost is also expected to improve. If we consider the GPT-family of models as an example, the cost of GPT-4 was initially of $0.03 and $0.06 per 1k input and output tokens respectively. However, GPT-4o-mini currently shows, being only slightly below GPT-4o in benchmarks, for a price more than 100 times cheaper than GPT-4.

Speed-wise, many efforts are put by academia and industry in designing new hardware (such as new GPUs), as well as in strategies for LLMs to run quicker on these pieces of hardware (e.g., FlashAttention (Dao et al., 2022), AWQ (Lin et al., 2024), etc.). It is therefore expected that LLMs will increase in speed over time, making the gap between workers and LLMs larger and larger.

5.4 Nature of Disagreements between LLMs and between People

We observed during our experiments that the nature of disagreements happens to be different for LLMs and people. A typical example of that is Twitter handles (such as @JohnDoe). During all our experiments, people kept struggling with Twitter handles. Understandably, it is not clear, in the original WNUT-17 guidelines, if using @JohnDoe at the beginning of a Tweet to indicate that the message is about John Doe makes @JohnDoe a "person" entity.

However, LLMs seem to generally agree on the fact that Twitter handles are not entities with the original WNUT-17 guidelines. Examples of LLM rationales for not annotating Twitter handles are provided in Appendix E.

This difference between people and LLMs makes that Twitter handles are always one the first things to clarify, for people, in the guidelines. For LLMs, however, it is something to clarify in a later stage. For instance, while the first iteration did not contain references to Twitter handles (see Table 3 in Appendix B), it is only at the second iteration that LLMs consider it worth it to mention them.

5.5 On Subjective Annotation Tasks

In our case study, we assumed that multiple interpretations of an annotation indicated an issue with either the annotation guidelines or the text being annotated. However, some annotation tasks are intrinsically subjective. For instance, annotating a piece of text as "well written" or not often depends on the perspective of the annotator. Two changes in our setup are needed to accommodate such a task.

First, the notion of agreement needs to be changed. Instead of checking if the annotators agree with each other, one may check if the distribution of annotations is expected. For instance, the percentage of administrative texts that are "understandable" (given a definition of what "understandable" means in the guidelines) should be close to an expected percentage given in the literature for a certain population.

Second, annotator LLMs should integrate personas such that the distribution of personas corresponds to the population simulated by the LLMs for the subjective annotation task.

6 Conclusion

In this paper, we presented a case study on automating complex annotation workflow. We provided some insights about using LLMs for the automation of such workflows. In particular, our case study suggests that LLMs can produce guidelines of good quality: from an inter-annotator agreement of 0.596 (original WNUT-17 guidelines) to 0.84 (LLM-improved guidelines). We also noted that the gap in cost and time required by workers and LLMs to go through the workflow was significantly large, with LLMs going more than 300 times quicker through the workflow, for a cost per annotator that is more than 700 times cheaper.

Based on our case study, we urge the community to develop LLM-specific workflows, as our case study seems to indicate that LLMs are well-suited for the task. However, further work is needed to identify the datasets and tasks for

which humans are superior than LLMs when going through annotation workflows. Thanks to that, a subsequent future work can be to categorize datasets and tasks for the community to better understand when to leverage crowdsource workers and when to develop LLM-based systems.

7 Limitations of our Case Study

One limitation of our study is that it is assessed on one dataset (WNUT-17) and one task (Entity Recognition) only. While it is true that multiplying the datasets and tasks would strengthen our conclusions, we believe that our case study is enough to convey some insights about the use of LLMs to automate complex annotation workflows. Furthermore, we also believe that cost and time-wise, the gap is so large that it is very unlikely to be disproved by analyzing many datasets and tasks.

However, an interesting future work would be to find particular datasets and tasks for which our conclusions would not hold. In particular, this means finding a dataset and a task for which people are a lot superior when going through annotation workflows than LLMs.

Acknowledgments

No data collection or experimentation was conducted by Meta. The workflow automation described in this paper was based on work supported by the Air Force Research Laboratory under Contract No. FA8750-22-C-0511 and by the Army ASA(ALT) SBIR CCoE under Contract No. W51701-24-C-0127. The experiments with crowd workers described here were carried out as part of InferLink's commercial activities. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funders.

References

Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. 2010. Soylent: A word processor with a crowd inside. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*, pages 313–322.

Jonathan Bragg, Mausam, and Daniel S Weld. 2018. Sprout: Crowd-powered task design for crowdsourcing. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*, pages 165–176.

Joseph Chee Chang, Saleema Amershi, and Ece Kamar. 2017. Revolt: Collaborative crowdsourcing for labeling machine learning datasets. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 2334–2346.

Quan Ze Chen and Amy X Zhang. 2023. Judgment Sieve: Reducing uncertainty in group judgments through interventions targeting ambiguity versus disagreement. *Proceedings of the ACM on Human-Computer Interaction (HCI)*, 7(CSCW2):1–26.

Quanze Chen, Jonathan Bragg, Lydia B Chilton, and Dan S Weld. 2019. Cicero: Multi-turn, contextual argumentation for accurate crowdsourcing. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–14.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:16344–16359.

Leon Derczynski, Eric Nichols, Marieke Van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 140–147.

Ryan Drapeau, Lydia Chilton, Jonathan Bragg, and Daniel Weld. 2016. Microtalk: Using argumentation to improve crowdsourcing accuracy. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 4, pages 32–41.

Marcio Fonseca and Shay B Cohen. 2023. Can large language models follow concept annotation guidelines? A case study on scientific and financial domains. *arXiv:2311.08704*.

Ujwal Gadiraju, Ricardo Kawase, Stefan Dietze, and Gianluca Demartini. 2015. Understanding malicious behavior in crowdsourcing platforms: The case of online surveys. In *Proceedings of the Annual ACM Conference on Human Factors in Computing Systems*, pages 1631–1640.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100.

Vivek Krishna Pradhan, Mike Schaekermann, and Matthew Lease. 2022. In search of ambiguity: A three-stage workflow design to clarify annotation guidelines for crowd workers. *Frontiers in Artificial Intelligence*, 5:828187.

James Pustejovsky and Amber Stubbs. 2012. *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications*. O'Reilly Media, Inc.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. GoLLIE: Annotation guidelines improve zero-shot information-extraction. In *International Conference on Learning Representations (ICLR)*.

Mike Schaekermann, Joslin Goh, Kate Larson, and Edith Law. 2018. Resolvable vs. irresolvable disagreement: A study on worker deliberation in crowd work. *Proceedings of the ACM on Human-Computer Interaction (HCI)*, 2(CSCW):1–19.

Manam VK Chaithanya, Dwarakanath Jampani, Mariam Zaim, Meng-Han Wu, and Alexander J. Quinn. 2019. Taskmate: A mechanism to improve the quality of instructions in crowdsourcing. In *Companion Proceedings of The World Wide Web Conference*, pages 1121–1130.

Manam VK Chaithanya and Alexander J. Quinn. 2018. Wingit: Efficient refinement of unclear task instructions. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 6, pages 108–116.

Meihong Wang, Yuling Sun, Jing Yang, and Liang He. 2018. Enabling the disagreement among crowds: A collaborative crowdsourcing framework. In *IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 790–795.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:24824–24837.

Huichuan Xia. 2024. Tragedy of the commons in crowd work-based research. *ACM Journal on Responsible Computing*, 1(1):4:1–4:25.

A Prompts Used in the LLM-based Solution

The LLM-based solution is built upon three components: annotator LLMs, an ensemble of these annotator LLMs and a guideline optimizer LLM. The prompt used for the annotator LLMs is the following:

> You are an expert in annotating entities in texts. You will be provided with annotation guidelines and, based on them, you will have to annotate the entities in a sentence. Here are the guidelines about the entities to annotate: [GUIDELINES]
>
> Each entity has two versions (B- and I-) depending on if a token starts the entity (B-) or not (I-). For instance, in the sentence "I'm Chuck Norris", "Chuck" should be annotated as B-person because "Chuck" starts the entity "Chuck Norris" and "Norris" should be annotated as I-person because "Norris" doesn't start the entity "Chuck Norris".
>
> The list of all entities available for annotation is therefore the following: [LIST OF POSSIBLE ANNOTATION CATEGORIES]
>
> Given the guidelines, what is the entity type of each of the following tokens [TOKENS IN THE SENTENCE TO ANNOTATE] in "[SENTENCE TO ANNOTATE]"? Think step-by-step and answer with a JSON containing two keys: (1) "reasoning", which will contain a list with your reasoning for each annotation, and (2) "annotation", which will contain your annotation only. Your annotation in the "annotation" key of the JSON must contain a list of entities in the format [O,O,B-person,I-person,I-person,O,B-location].

where all elements in brackets are elements to be provided in the prompt.

For the guideline optimizer LLM, the prompt is the following:

> You are an expert in making annotation guidelines better. You will be provided with annotation guidelines and some elements on which there is some disagreements. Your goal is to improve the provided guidelines to reduce the disagreement between the annotators. Here are the annotation guidelines to improve: "[CURRENT GUIDELINES]"
>
> These guidelines currently have an inter-annotator agreement of [INTER-ANNOTATOR AGREEMENT]. The disagreement is mainly because of disagreements between these elements: [EXAMPLES OF DISAGREEMENT]

> Provide a new version of these guidelines in order to solve these disagreements. When updating the guidelines, make them so that there will not be new disagreements on the following sentences that will be annotated: [SENTENCES USED IN THE ANNOTATION PROCESS]
>
> The only thing you can do is to clarify the description of categories in order to reduce future disagreements. In other words, only change the description of person, location, corporation, product, creative work and group. If you want to provide examples in the descriptions about what to do or not do, invent examples, i.e. do not take examples from the dataset. Finally, do not mention the B and I of the categories in the description (e.g., B-group and I-group). Instead, mention the category itself (e.g., group).
>
> Answer with nothing else but a string corresponding to the new guidelines you propose.

where all elements in brackets are elements to be provided in the prompt.

B Examples of Guideline Improvements

Tables 2 and 3 show comparisons of the original WNUT-17 guidelines with the worker-based improvements after one iteration (Table 2) and with the LLM-based improvements after one iteration (Table 3).

C Experimental Guidelines

Table 4 shows the resulting guidelines after four iterations over the LLM-based workflow. The amount of agreements resulting from these guidelines has been assessed in Section 4.3.1, with the results reported in Table 1.

D Cost and Time Insights

Table 5 reports the time taken by workers to go through each phase of the workflow. 16, 10, 8 and 12 workers went through Phases 1, 2, 3 and then Phase 1 again in the workflow. Going through the workflow once, and then annotating again, spanned roughly one week and a half. This is due to several factors. First of all, all workers did not start at the same time – a worker can hold onto a sheet for 30 minutes, then can decide that they do not want to work on it, releasing the sheet for another worker 30 minutes after the first ones started. However, this accounts for only short delays. Most important delays are due to the fact that some phases are connected (e.g., Phase 2 and Phase 1, as Phase 2 is about asking for rationales related to annotations in Phase 1). Because of that, the annotation manager had to wait until the workers from Phase 1 were available again to do the second phase. Lastly, many workers had issues with the platform, and a significant number of them cheated (tried to get the completion code, in order to be paid, without doing the task), did not do the task in its entirety or did it in a seemingly random way. Because of that, a significant amount of time of the annotation manager was required to handle these issues. Section 5.1 elaborates on that.

In comparison, going through the workflow once, and then annotating again, is performed in 18.43 seconds by the LLMs (see Table 6). Indeed, the average runtime to perform the annotation with the initial guidelines is 5.83 seconds per LLM per instance/sentence. As all the calls to the LLMs are parallelized, having 10 annotator LLMs and 20 sentences to annotate still is 5.83 seconds in total on average. Runtime of going through the workflow once and then annotating is therefore the sum of the runtime for the annotation (which is parallelized), the optimization of the guidelines (only one call to an LLM) and the re-annotation (which is also parallelized).

Cost-wise, the cost of an LLM is generally computed in two different ways, depending on the situation: either the LLM is self hosted, or it is accessed via an API. In the first case, the cost of the LLM is the cost of the infrastructure used to work with the LLM. For example, if Amazon AWS is used with a specific instance (e.g., an ml.t3.medium), then the cost per hour of this instance multiplied by the time needed to complete the workflow will define the cost. If the model is not self hosted, but rather accessed via an API, then the cost is generally dependent on the number of tokens in the input and output (with input and output tokens varying in price).

In our case, as GPT-4o was used in this study, the cost is per token. The number of input and output tokens needed for each phase, as well as the corresponding costs, are presented in Table 6. The average number of input and output tokens for 10

Initial guidelines (before any iteration)	Worker-based solution after 1 iteration
• person: Names of people (e.g. Virginia Wade). Don't mark people that don't have their own name. Include punctuation in the middle of names. Fictional people can be included, as long as they're referred to by name (e.g. Harry Potter).	• person: Names of people (e.g. Virginia Wade). Don't mark people that don't have their own name. Include punctuation in the middle of names. Fictional people can be included, as long as they're referred to by name (e.g. Harry Potter). Usernames are considered a name to identify a person (e.g. @JohnDoe).
• location: Names that are locations (e.g. France). Don't mark locations that don't have their own name. Include punctuation in the middle of names. Fictional locations can be included, as long as they're referred to by name (e.g. Hogwarts).	• location: Names that are locations (e.g. France). Don't mark locations that don't have their own name. Include punctuation in the middle of names. Fictional locations can be included, as long as they're referred to by name (e.g. Hogwarts). Twitter handles about locations should be considered as locations.
• corporation: Names of corporations (e.g. Google). Don't mark locations that don't have their own name. Include punctuation in the middle of names.	• corporation: Names of corporations (e.g. Google). Don't mark locations that don't have their own name. Include punctuation in the middle of names. Twitter handles for corporations should be considered as corporations.
• product: Name of products (e.g. iPhone). Don't mark products that don't have their own name. Include punctuation in the middle of names. Fictional products can be included, as long as they're referred to by name (e.g. Everlasting Gobstopper). It's got to be something you can touch, and it's got to be the official name.	• product: Name of products (e.g. iPhone). Don't mark products that don't have their own name. Include punctuation in the middle of names. Fictional products can be included, as long as they're referred to by name (e.g. Everlasting Gobstopper). It's got to be something you can touch, and it's got to be the official name.
• creative_work: Names of creative works (e.g. Bohemian Rhapsody). Include punctuation in the middle of names. The work should be created by a human, and referred to by its specific name.	• creative_work: Names of creative works (e.g. Bohemian Rhapsody). Include punctuation in the middle of names. The work should be created by a human, and referred to by its specific name.
• group: Names of groups (e.g. Nirvana, San Diego Padres). Don't mark groups that don't have a specific, unique name, or companies (which should be marked corporation).	• group: Names of groups (e.g. Nirvana, San Diego Padres). Don't mark groups that don't have a specific, unique name (e.g. "a group of runners" is not a specific name, as opposed to "Los Angeles Runners"), or companies (which should be marked as corporations). Twitter handles can be considered as specific, unique names for groups.

Table 2: Guidelines improved after 1 iteration over the workflow for the worker-based solution. The first column of the table shows the initial guidelines. Highlighted in green are the changes made by the solution.

annotators and 20 sentences in the table are computed by taking the average for 1 annotator LLM annotating 1 sentence, multiplied by 200 (for the 200 calls corresponding to 10 annotator LLMs annotating 20 sentences). The result is an approximate cost of $4 to go through the complex workflow with 10 annotator LLMs and 20 sentences.

Concerning crowd workers, the cost related to workers depends on two variables: the time a specific phase takes for them and the rate per hour paid to the workers. The time needed for workers to go through each phase of the workflow is reported in Table 5. It can be seen that, in addition to being significantly slower, as each phase takes roughly 30 minutes per worker, the cost of each phase per worker is around $4, $5 and $10 for an hourly rate of $8, $10 and $20 respectively ($8 per hour being the minimum on Prolific). Therefore, at the cost of having one worker going through one phase of the workflow with 20 sentences, while

Initial guidelines (before any iteration)	LLM-based solution after 1 iteration
• person: Names of people (e.g. Virginia Wade). Don't mark people that don't have their own name. Include punctuation in the middle of names. Fictional people can be included, as long as they're referred to by name (e.g. Harry Potter). • location: Names that are locations (e.g. France). Don't mark locations that don't have their own name. Include punctuation in the middle of names. Fictional locations can be included, as long as they're referred to by name (e.g. Hogwarts). • corporation: Names of corporations (e.g. Google). Don't mark locations that don't have their own name. Include punctuation in the middle of names. • product: Name of products (e.g. iPhone). Don't mark products that don't have their own name. Include punctuation in the middle of names. Fictional products can be included, as long as they're referred to by name (e.g. Everlasting Gobstopper). It's got to be something you can touch, and it's got to be the official name. • creative_work: Names of creative works (e.g. Bohemian Rhapsody). Include punctuation in the middle of names. The work should be created by a human,and referred to by its specific name. • group: Names of groups (e.g. Nirvana, San Diego Padres). Don't mark groups that don't have a specific, unique name, or companies (which should be marked corporation).	• person: Names of people, including titles or roles when they are part of a name (e.g., President Lincoln). Do not mark generic titles or roles without a specific name. Include punctuation in the middle of names. Fictional people can be included, as long as they're referred to by name (e.g., Harry Potter). • location: Names that are specific geographic locations or landmarks (e.g., Eiffel Tower, France). Do not mark generic terms like "park" or "river" without a specific name. Include punctuation in the middle of names. Fictional locations can be included, as long as they're referred to by name (e.g., Hogwarts). • corporation: Names of corporations or companies (e.g., Google, Microsoft). Do not mark generic terms like "store" or "company" without a specific name. Include punctuation in the middle of names. • product: Names of tangible products or items (e.g., iPhone, Coca-Cola). Do not mark services or intangible products. Include punctuation in the middle of names. Fictional products can be included, as long as they're referred to by name (e.g., Everlasting Gobstopper). It must be something you can touch, and it must be the official name. • creative_work: Names of creative works such as books, songs, movies, or artworks (e.g., Bohemian Rhapsody, The Great Gatsby). Include punctuation in the middle of names. The work should be created by a human, and referred to by its specific name. • group: Names of groups, teams, or bands (e.g., Nirvana, San Diego Padres). Do not mark generic terms like "team" or "band" without a specific name. Do not mark companies, which should be marked as corporation.

Table 3: Guidelines improved after 1 iteration over the workflow for the LLM-based solution. The first column of the table shows the initial guidelines. Green highlights correspond to elements added by the LLM optimizer, red ones correspond to what has been deleted, and orange ones to reformulations.

being paid the minimal hourly rate on Prolific, one can have 10 annotator LLMs doing an entire iteration over the workflow.

E LLM Rationales for not Annotating Twitter Handles

During our experiments, we could see that knowing if a Twitter handles was an entity or not was an important issues for workers. However, this was generally not a primary concern for LLMs, for which clarifying Twitter handles came at a later stage of the process. Table 7 shows some examples of LLM rationales for not annotating Twitter handles.

Initial guidelines (before any iteration)	LLM-based solution after 4 iterations
• person: Names of people (e.g. Virginia Wade). Don't mark people that don't have their own name. Include punctuation in the middle of names. Fictional people can be included, as long as they're referred to by name (e.g. Harry Potter).	• person: Names of people, including titles or roles when they are part of a name (e.g., President Lincoln). Do not mark generic titles or roles without a specific name. Include punctuation in the middle of names. Fictional people can be included, as long as they're referred to by name (e.g., Harry Potter). Social media handles or usernames that clearly refer to a person should also be marked as person. Do not mark possessive forms or contractions as person unless they are part of a name. Do not mark standalone words or punctuation that are not part of a person's name.
• location: Names that are locations (e.g. France). Don't mark locations that don't have their own name. Include punctuation in the middle of names. Fictional locations can be included, as long as they're referred to by name (e.g. Hogwarts).	• location: Names that are specific geographic locations or landmarks (e.g., Eiffel Tower, France). Do not mark generic terms like "park" or "river" without a specific name. Include punctuation in the middle of names. Fictional locations can be included, as long as they're referred to by name (e.g., Hogwarts). Do not mark dates, days of the week, or months as locations. Do not mark standalone words or punctuation that are not part of a location's name.
• corporation: Names of corporations (e.g. Google). Don't mark locations that don't have their own name. Include punctuation in the middle of names.	• corporation: Names of corporations or companies (e.g., Google, Microsoft). Do not mark generic terms like "store" or "company" without a specific name. Include punctuation in the middle of names. Do not mark groups, teams, or bands as corporations. Do not mark standalone words or punctuation that are not part of a corporation's name.
• product: Name of products (e.g. iPhone). Don't mark products that don't have their own name. Include punctuation in the middle of names. Fictional products can be included, as long as they're referred to by name (e.g. Everlasting Gobstopper). It's got to be something you can touch, and it's got to be the official name.	• product: Names of tangible products or items (e.g., iPhone, Coca-Cola). Do not mark services or intangible products. Include punctuation in the middle of names. Fictional products can be included, as long as they're referred to by name (e.g., Everlasting Gobstopper). It must be something you can touch, and it must be the official name. Do not mark generic terms like "truck" or "car" unless they are part of a specific product name. Do not mark verbs, actions, standalone words, or punctuation related to products.
• creative_work: Names of creative works (e.g. Bohemian Rhapsody). Include punctuation in the middle of names. The work should be created by a human,and referred to by its specific name.	• creative_work: Names of creative works such as books, songs, movies, or artworks (e.g., Bohemian Rhapsody, The Great Gatsby). Include punctuation in the middle of names. The work should be created by a human, and referred to by its specific name. Do not mark parts of a date or time as creative works. Do not mark conjunctions, prepositions, standalone words, or punctuation as part of creative works unless they are part of the official title.
• group: Names of groups (e.g. Nirvana, San Diego Padres). Don't mark groups that don't have a specific, unique name, or companies (which should be marked corporation).	• group: Names of groups, teams, or bands (e.g., Nirvana, San Diego Padres). Do not mark generic terms like "team" or "band" without a specific name. Do not mark companies, which should be marked as corporation. Social media handles or usernames that clearly refer to a group should also be marked as group. Do not mark standalone words or punctuation that are not part of the group's name.

Table 4: Guidelines improved after 4 iterations over the workflow for the LLM-based solution. The first column of the table shows the initial guidelines. Green highlights correspond to elements added by the LLM optimizer, red ones correspond to what has been deleted, and orange ones to reformulations.

Phase	Median time needed for the phase (with min and max)
Phase 1 (Annotation)	28:54 minutes (15:47 minutes - 53:23 minutes)
Phase 2 (Justify)	25:10 minutes (8:29 minutes - 55:30 minutes)
Phase 3 (Reconsider)	23:12 minutes (12:46 minutes - 1:02:52 hour)
Phase 1 (Annotation w/ new guidelines)	34:54 minutes (14:32 minutes - 49:50 minutes)
Total	1:52:10 hour (51:12 minutes - 3:41:35 hours)

Table 5: Time required by the workers to go through each phase of the workflow. The number of workers who went through each of these phases is 16, 10, 8 and 12 for, respectively, Phases 1, 2, 3 and then Phase 1 again.

Phase	Avg # input tokens per LLM	Avg # output tokens per LLM	Cost	Avg runtime
Annotation + Justify	771	499	$0.0069175	5.83 sec
Modifying the guidelines	1871	472	$0.0093975	5.9 sec
Annotation w/ new guidelines	784	503	$0.00699	6.7 sec
Total for 1 annotator and 1 sentence	3429	1174	$0.0203125	18.43 sec
Total for 10 annotators and 20 sentences	685,800	234,800	$4.0625	18.43 sec

Table 6: Number of input and output tokens needed for each phase per annotator LLM, and the corresponding cost and runtime for GPT-4o. The information is reported for each instance/sentence on average ("# input tokens" and "# out tokens"). At the time of this analysis, the cost of GPT-4o was $0.0025/1K input tokens and $0.01/1K output tokens. The input prompt for the annotation includes the instructions for the task, the guidelines of WNUT17 and the sentence to annotate. The average runtime from multiple annotators and sentences do not scale because all the annotations are done in parallel. Please note that for speed and accuracy reasons, Phase 1 and 2 are mixed for LLMs, and because the results are mostly similar before and after Phase 3 (see Section 3.2), Phase 3 is skipped in the automation of the workflow.

Entity	Sentence	LLM's Rationale
@TeenThings	RT @TeenThings : I love waking up in the middle of the night knowing i have more time to sleep. #teenthings #comedown-withlove	This seems to be a Twitter handle, which doesn't correspond to named entities like person, location, etc. in the context of physical entities, so it's labeled as 'O'. --- This is a Twitter handle. While it typically represents a user or group, it doesn't fit neatly into the provided categories like person, group, or corporation. As handles aren't named entities per se, it will also be annotated as 'O'. --- Appears to be a social media username. It doesn't fit any provided entity categories (person, location, corporation, etc.). Annotated as O.
@KaulitzClub	@KaulitzClub Hahaha! We dance, eat, drink, and talk EVERYTIME we get together. And it's been this way for like years now!	This appears to be a social media handle or username rather than a name of a person, location, corporation, product, creative work, or group. Thus, it should be marked as 'O'. --- It appears to be a username or a handle, but it doesn't fit into any of the specified annotation categories, such as person or group. Thus, it is marked as O. --- This could be a group or corporation, but based on the given format, it's a social media handle. Without specific context identifying it as a named entity, it's marked as O.

Table 7: Examples of LLM rationales for *not* annotating Twitter handles with the original WNUT-17 guidelines. Please note that 'O' is used by the models to say that it is not an entity.