Human-LLM Collaborative Annotation Through Effective Verification of LLM Labels

Xinru Wang*
xinruw@purdue.edu
Purdue University
West Lafayette, IN, USA

Hannah Kim
hannah@megagon.ai
Megagon Labs
Mountain View, CA, USA

Sajjadur Rahman
sajjadur@megagon.ai
Megagon Labs
Mountain View, CA, USA

Kushan Mitra
kushan@megagon.ai
Megagon Labs
Mountain View, CA, USA

Zhengjie Miao*
zhengjie@sfu.ca
Simon Fraser University
Burnaby, BC, Canada

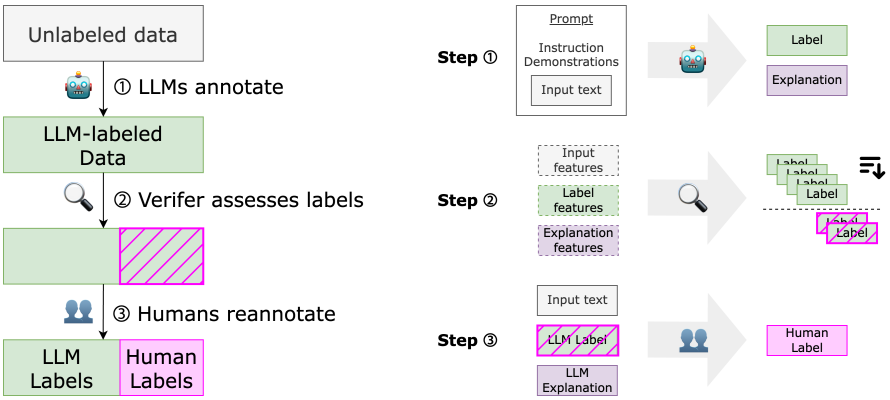


Figure 1: (Left) Our human-LLM collaborative annotation framework. (Right) Inputs and outputs of each step in our framework. Step 1: LLMs predict labels and generate explanations. Step 2: Verifier assesses LLM labels and explanations. Step 3: Human annotators re-annotate instances with lowest verifier scores.

ABSTRACT

Large language models (LLMs) have shown remarkable performance across various natural language processing (NLP) tasks, indicating their significant potential as data annotators. Although LLM-generated annotations are more cost-effective and efficient to obtain, they are often erroneous for complex or domain-specific tasks and may introduce bias when compared to human annotations. Therefore, instead of completely replacing human annotators with LLMs, we need to leverage the strengths of both LLMs and humans to ensure the accuracy and reliability of annotations. This paper presents a multi-step human-LLM collaborative approach where (1) LLMs generate labels and provide explanations, (2) a verifier assesses the quality of LLM-generated labels, and (3) human annotators re-annotate a subset of labels with lower verification scores. To facilitate human-LLM collaboration, we make use of LLM's ability to rationalize its decisions. LLM-generated explanations can provide additional information to the verifier model as well as help humans better understand LLM labels. We demonstrate that our verifier is able to identify potentially incorrect LLM labels for human re-annotation. Furthermore, we investigate the impact of presenting LLM labels and explanations on human re-annotation through crowdsourced studies.

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in HCI**; • **Computing methodologies** → **Machine learning**; *Natural language generation.*

KEYWORDS

Human-LLM collaborative annotation, LLM annotation, self-rationalization, text annotation, NLP

ACM Reference Format:
Xinru Wang, Hannah Kim, Sajjadur Rahman, Kushan Mitra, and Zhengjie Miao. 2024. Human-LLM Collaborative Annotation Through Effective Verification of LLM Labels. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 21 pages. https://doi.org/10.1145/3613904.3641960

*Work done at Megagon Labs.

CHI '24, May 11–16, 2024, Honolulu, HI, USA

ACM ISBN 979-8-4007-0330-0/24/05
https://doi.org/10.1145/3613904.3641960

1 INTRODUCTION

Recent large language models (LLMs) have demonstrated significant progress in their rich abstraction capabilities to understand and generate text. These models, such as OpenAI's GPT series [2, 9, 66], and Google PaLM [4], which learn to predict the next word based on the sequence of words that they already saw, implicitly represent a large amount of general knowledge from the training corpus in their billions of parameters, which enables them to solve various tasks like translation, summarization, conversation, and reasoning by generating outputs following the input text prompts.

While LLMs have excited the world with superior performance in these generation-based tasks, they have also emerged as data annotators through their advanced semantic understanding abilities. Unlike previous deep learning based solutions where one model only works for a fixed pre-defined set of labels, LLMs provide unprecedented flexibility such that users can feed them with customized text prompts for any annotation task. Recent works have investigated adapting LLMs such as GPT-3 [90] and ChatGPT [25, 30, 41, 83] to annotate data for text classification and natural language generation tasks. While a fully automated data annotation process would free human annotators and significantly reduce annotation costs, state-of-the-art LLMs are not a panacea: empirical results indicate that the performance of LLMs substantially depends on tasks, datasets, and labels [107, 108], and thus they cannot replace human annotators. For some popular classification tasks, human annotators can still outperform LLMs consistently as shown in Table 1.

Therefore, we need to leverage the strengths of both LLMs and humans to ensure the accuracy and trustworthiness of annotations. There are two lines of work in recent LLM research that can help us achieve the best of two worlds. The first one is to leverage the LLM's capability to generate fluent and supportive explanations for its predictions that help rationalize its reasoning process [6, 46, 59, 73, 89, 98, 99], hence to assist humans to understand and trust model predictions [87]. The second line of work studies how to verify the outputs of LLMs. For more complex tasks like math reasoning [12, 47, 76], commonsense reasoning [47], and semantic parsing [64], previous works train a separate model (usually a smaller language model) to judge the correctness of LLM-generated responses. For example, Li et al. [47] propose to ask the LLM to generate multiple reasoning paths with diverse prompts, and then they train a smaller language model based verifier to score the quality of each reasoning path that later leads to the aggregate score of the prediction.

Inspired by these two lines of work, we design a multi-step human-LLM collaborative framework for data annotation where these two lines of work integrate perfectly. In this framework, we first use LLMs to generate labels and provide explanations for their label prediction on unlabeled data. Then, we train a separate simple verifier model to assess the quality of LLM's annotation, with the help of pre-extracted textual features from the example-to-annotate, the corresponding explanations, and LLM's confidence about its prediction. Finally, human annotators re-annotate a subset of labels selected by the verifier according to the quality score. Note that LLM-generated explanations are used in both the verification and re-annotation steps: we expect those explanations would provide more information on how the LLM makes the prediction or even demonstrate defects that the verifier can use to distinguish unreliable annotations; moreover, as shown in previous studies [18, 36, 38], explanations can improve human-AI collaboration in terms of trust and annotation performance. To build such a framework that synergies human and LLM annotation through explanation and verification, in this paper, we explore the following research questions:

- **RQ1:** How to verify LLM-generated labels using signals from inputs, LLM labels, and LLM explanations?
- **RQ2:** Does providing LLM-generated labels and explanations help humans in re-annotation?

We name our framework as Lapras (LAnguage model Powered Reliable Annotation aSsistant) and empirically evaluate it on six text classification benchmarks and crowdsourced studies. Our results show that the random forest based verifier trained on only 200 labeled data can achieve much higher accuracy than baseline methods that use LLM uncertainty metrics. Our analysis of crowd worker annotations indicates the importance of the quality of LLM-generated explanations to their preference for AI assistance and annotation performance. Following our findings, we discuss answers to the above research questions on improving the LLM annotation verifier and enhancing human-AI collaboration in the re-annotation step. We discuss the limitation of alternative verification methods for LLM outputs, possible improvements on the re-annotation interface to present LLM explanations, and the need for prompt-tuning in the human-LLM collaborative annotation. Our findings also provide important implications from the human-subject study: improving the quality of LLM-generated explanations and deciding when it is helpful to show the explanations to humans are promising directions, potentially boosting the effectiveness of human-LLM collaborative annotation frameworks. Nevertheless, our findings and discussions may be limited by the performance of crowdsourced annotations and the way we collect gold labels for training the verifier. Despite these limitations, we are optimistic about springing future works in developing human-LLM collaborative annotation and better support for LLM verification with self-explanation.

2 RELATED WORK

2.1 Large Language Models and Prompting

Large language models (LLMs) generate response text based on a textual input, which is called a prompt. Prompts guide LLM's response generation process and usually consist of task instruction and demonstration examples. Based on a given prompt, LLMs can follow task instructions, learn from examples, and generate human-like responses. A plethora of approaches to improve LLM prompting has been introduced recently, as listed in survey papers [17, 55]. Note that prompt tuning or prompt engineering techniques are out of the scope of our paper, but they can be applied to our annotation setting to enhance the performance of LLMs. In the following, we focus on LLM's ability to rationalize its responses as well as approaches to assess LLM's responses.

2.1.1 Self-rationalization of LLMs. Given a data label, LLMs have shown capabilities to generate fluent and supportive explanations

Table 1: LLM and human accuracy on some common tasks. Numbers without reference are from our own evaluation on 2,000 samples.

Model	Textual Entailment (SNLI)	Textual Entailment (MNLI)	Stance Detection (SemEval)	Semantic Change (WiC)	Hate Speech Detection (HateXplain)
GPT 3.5 (zero-shot)	0.62	0.69	0.41	0.6 [30]	-
GPT 3.5 (3-shot)	0.79	0.78	0.67	0.66 (8-shot) [30]	0.6
Human accuracy	0.97 [23]	0.92 [63]	0.82 [33]	0.8 [30]	0.81 [60]

for the selected label [73, 88], which are often preferred over human-generated explanations [98]. Self-rationalization models, beyond merely generating rationales for an existing label, possess the capability to make new predictions and provide explanations for the predicted labels simultaneously. Several works [6, 59, 89, 99] investigate self-rationalization capabilities of LLMs for various tasks including classification and question answering. Challenges remain on whether LLM-generated explanations are faithful to their underlying decision process. For example, Turpin et al. [85] demonstrates that chain-of-thought explanations can be influenced by biased context. In this work, we ask LLM to label and explain (self-rationalize) similarly to [41, 107], in order to better understand the LLM's labeling decision process. We conjecture that signals from LLM explanations can help in the assessment of the accuracy of the corresponding LLM labels.

2.1.2 Validation of LLM responses. Despite the strong reasoning and generalization capabilities across a wide range of NLP tasks, LLMs are deficient in terms of error detection or correction mechanisms. Hence, scoring, filtering, or re-ranking LLM outputs have been common for math reasoning [12, 47, 76], commonsense reasoning [47], and semantic parsing [64], by training another smaller language model. Previous works have also studied the process of LLMs self-verifying their output, either by asking the models to self-evaluate their answers [35] or by measuring the uncertainty or confidence score associated with the answers [50, 69, 93, 101]. In this paper, we target text classification tasks and demonstrate that in addition to output signals, including explanations and characteristics of the input corpus will improve the validation performance.

2.2 AI-Assisted Annotation

Rapid advancements in machine learning (ML) heavily rely on the availability of abundant annotated training data, requiring a costly and time-consuming process of data labeling. One way to streamline data annotation is combining automatic pre-annotation with manual annotation to improve efficiency while maintaining annotation quality. In the pre-annotation phase, data are automatically labeled by an existing ML system, with human annotators then correcting any inaccuracies [77]. Studies have found that highly accurate pre-annotation significantly reduces annotation time [20, 51, 61] and enhances the consistency of annotations without compromising quality [61]. On the opposite, poor-quality pre-annotations can produce little improvement in efficiency or data quality [65, 79].

To further reduce cost, it is crucial to select the most informative instances for human annotation. Active learning (AL) techniques, like PAL (Pre-annotation and Active Learning) [78] offer a solution for facilitating the selection of pre-annotation data needed. Ringger et al. [71] present simple linear models to estimate the cost of annotation, with application in active learning. Wang et al. [86] also find that informative AL contributes to this cost reduction effort.

Numerous annotation tools have been developed to facilitate the use of pre-annotation and empower human annotators in tasks spanning natural language [61, 82], image segmentation [3], and various application domains such as clinical text labeling [31, 40, 79] and seismic data labeling [24]. Gebreegziabher et al. [21] develop a tool for qualitative coding, a type of under-explored annotation task. The Fluid Annotation introduced by Andriluka et al. [3] enables putting human effort only on the errors in pre-annotation. Hernandez et al. [31] showcase the effectiveness of the machine assistance in their annotation system in improving non-experts' annotation performance in drug product labeling.

2.2.1 LLMs as annotators. Recently, researchers have started investigating the potential of LLMs as data annotators to reduce the cost and effort needed to collect training data. Wang et al. [90] are among the early adopters of utilizing GPT-3 for the purpose of annotating data for classification and natural language generation tasks. Their results indicate that GPT annotation can significantly reduce annotation costs (up to 96%), but human labeling still outperforms LLMs given enough budget. This paper also proposes an active labeling strategy where humans re-annotate data instances with the lowest confidence scores, which motivated our work. Ding et al. [16] also uses GPT-3 as an annotator and conducts comprehensive experiments on classification and token-level tasks. Several works [41, 107, 108] investigate the labeling capabilities of LLMs for social science tasks such as stance detection, hate speech detection, narrative analysis, and so on. Zhu et al. [107], Ziems et al. [108] find that although LLMs are powerful, they cannot replace human annotators since performance varies substantially depending on tasks, datasets, and labels. Gilardi et al. [25], He et al. [30], Törnberg [83] compare the performance of ChatGPT with crowd workers and show that ChatGPT outperforms crowd workers on several classification tasks.

Another approach to leverage LLMs to prepare training data is data augmentation. Given a task and a label, LLMs can generate input text that corresponds to the given label [16, 74]. For example, Ding et al. [16] reports that data generation by GPT is more cost-effective than data annotation by GPT. In this paper, we only consider data annotation and leave human-LLM collaborative data augmentation as future work.

LLM-based annotation assistance differs from traditional ML-assisted annotation in several ways. First, ML models employed in traditional annotation assistance require task-specific training. For example, a topic classification model is not suitable for sentiment detection annotation. In contrast, pre-trained LLMs are easily

generalizable to a variety of tasks through prompt adjustments. Another difference is the observability of model confidence. Unlike most ML models, where uncertainty levels can be explicitly calculated, the uncertainty of LLMs (and most deep neural networks) is not observable. Instead, LLM confidences are approximated using logits, self-consistency [11], or self-evaluation [35] methods, and it turns out that LLMs are often over-confident [101], indicating a significant need for model calibration techniques [27] to improve the confidence estimates. In addition, for the models that provide explanations to aid in labeling decisions, the format of explanation may be different. While traditional ML models often produce feature-based or extractive explanations, LLMs can generate natural language explanations. Lastly, previous ML-assisted annotation work often utilizes highly accurate models. In this work, we do not assume that LLMs are flawless and acknowledge that their performance can vary across different tasks and datasets. Therefore, insights derived from previous research on ML-based annotation assistance cannot be directly transferred to our LLM-assisted setting, given the unique benefits and challenges introduced by hiring LLMs as annotators.

2.3 Empirical Studies in Human-AI Collaborative Systems

A number of empirical studies have been carried out to investigate the factors influencing human decision-making performance in human-AI collaborative systems. Most of these studies have concluded that the inclusion of a model decision improves performance compared to when no model decision is provided [26, 37, 42]. Performance is also improved when information about model confidence is presented [56, 92]. Another factor affecting human's model adoption is the accuracy of a model (both stated accuracy and observed accuracy) [42, 81, 103]. The effect of adding a model explanation on people's performance has been inconsistent across studies [81, 91]. For instance, Zhang et al. [106] found that displaying confidence scores and local explanations does not necessarily enhance AI-assisted decision-making. Similarly, Bansal et al. [5] found no significant improvement in human-AI team performance when explanations were communicated, compared to only showing the AI's confidence. Conversely, other studies have found that the effectiveness of AI explanations on human performance may largely be dependent on factors such as the explanation formats [42, 53, 94, 95] and the meaningfulness of the explanations to humans [96].

3 HUMAN-LLM COLLABORATIVE ANNOTATION FRAMEWORK

As LLMs such as OpenAI GPT series [9, 66], PaLM 2 [4], and LLaMA [84] demonstrate remarkable performance in various downstream tasks, there has been a recent surge in interest regarding collaborative data annotation involving both LLMs and humans. Recently, people have proposed pipelines where LLMs generate labels to augment human labels for training small downstream models [30, 90] or even directly replacing crowdsourced annotators [25, 107]. While the latest LLMs perform much better than humans for some specific tasks, in some cases, LLMs still fail to achieve superior accuracy [108] or fail to understand certain domain knowledge. Therefore, LLM annotation and human annotation would be a natural synergy that complements each other [90, 104]. In Table 1, we list the accuracy of GPT 3.5 and human annotators of five popular datasets for natural language inference, stance detection, semantic change, and hate speech detection. Although the accuracy improves a lot with in-context learning, GPT 3.5 still cannot outperform human annotators for these tasks.

Moreover, many tasks such as hate speech detection require the annotation to be trustworthy [87], while the prediction of LLMs is not transparent and may even further introduce biased or toxic predictions [49], having humans involved would further enhance explainable AI solutions and provide quality control over the annotations [52, 104].

To this end, we propose an innovative multi-step framework for collaborative annotation involving both LLMs and human annotators, aimed at reducing cost and enhancing efficiency and trustworthiness (depicted in Figure 1). The proposed LAPRAS framework first uses an LLM to annotate unlabeled data, exploiting the lower cost and scalability of LLMs; then, it ranks the annotations using a verifier, which selects a subset of potentially incorrect annotations that need further examination; finally, it distributes this potentially bad subset to human annotators for re-annotation. Below, we will demonstrate each step in LAPRAS.

3.1 Step 1: LLM Annotation

In this initial stage, the LLM generates labels for input samples and provides corresponding explanations justifying its decisions (Figure 1 (right-top)). LLMs exhibit great generalizability and work well as zero-shot and few-shot annotators [9]. In the simple zero-shot setting, the LLM input only consists of the task instructions, label space, and the text to annotate. To better leverage LLMs' ability, we also include few-shot demonstrations. Furthermore, recent studies have found that LLM-generated explanations would help explain its reasoning process, train smaller models [45, 46], and assist humans to understand and trust model predictions [87]. Therefore, in our framework, we ask the LLM to generate natural language explanations for their classification prediction, and then feed LLM-generated explanations to the verifier model (see Section 3.2) and also present them to human annotators (see Section 3.3).

The used task instructions are imperative sentences starting with verbs such as "Classify" and "Explain," outlining each LLM subtask. The number of demonstrations is determined by the number of classes in each dataset, with one demonstration for each class. Each demonstration comprises the data sample (i.e., the hypothesis and the premise sentences for natural language inference tasks, the tweets for stance detection tasks, and the social media post for hate speech detection tasks), the ground-truth answer, and a corresponding free-text explanation. We leverage the existing human-annotated explanations available in existing datasets or manually crafted explanations for the demonstration instances. The target input follows the same format as the demonstrations. The prompt ends with an "Answer:" prefix. In our evaluation with GPT 3.5, we observe that GPT consistently place its label result in the first sentence, followed by an explanation starting with "Explanation:"

on a separate line. We then processe GPT's responses to extract both classification labels and explanations.

3.2 Step 2: Verifier Assesses LLM Labels

After getting labels and explanations from LLM, it is ideal to examine each label and correct the wrong labels. However, having humans examine all LLM-generated labels would sacrifice the scalability and cost reduction of using LLMs for annotation. Hence, our subsequent task is to identify a subset of potentially erroneous LLM-generated labels for human re-annotation.

3.2.1 Annotation verification. In the literature, there are two major ways of measuring the uncertainty in LLM's output. One is to let the LLM self-evaluate or verbally express its certainty [50], and the other is to prompt the LLM multiple times with different prompt format to measure its consistency [93]. Although these measures are helpful, they may not work well for our settings due to (i) LLMs tend to be over-confident [101], (ii) the predicted label is pretty consistent in our observation since we only target classification tasks. Besides using LLM's own output to measure the uncertainty, people have shown that training a separate verifier to re-rank LLM's outputs is effective for reasoning and semantic parsing tasks [12, 47, 64]. Therefore, we develop a verifier model that "verifies" each LLM-generated label, which assigns scores to the quality of LLM-generated labels. The resulting scores help identify a subset of potentially incorrect LLM labels to be re-annotated by humans, which avoids wasting human effort on correcting already accurate labels.

Figure 1 (right-middle) shows the schematic representation of our verifier model's structure. Besides simply feeding the text and label to a smaller language model like BERT, we also extract and analyze informative signals from the data sample x, LLM label $\hat{y}$, and LLM explanation $\hat{e}$, as potential indicators to flag incorrect LLM labels that needed re-annotation. Those signals are then channeled into the verifier model, which learns a scoring function that measures how likely $\hat{y}$ is the correct output for input x. Hence, we denote the verifier as a binary classification model $\mathcal{P}_\theta(v|x, \hat{y}, \hat{e})$, where $v \in \{0, 1\}$. $v = 1$ indicates that the LLM annotation passing the verification (LLM label $\hat{y}$ is the same as the ground truth label). Then, we rank the examples according to the $\mathcal{P}_\theta(v|x, \hat{y}, \hat{e})$, and the returned result of this verification step are the bottom instances (with either a threshold or a fixed number of re-annotation) with lowest probability to pass the verification.

3.2.2 Verifier methods. **Training data.** As mentioned above, for text annotation tasks, each data sample consists of x, y^* where x is the text input and y^* is the gold label. For each data sample x, we use LLM to generate label $\hat{y}$ and corresponding explanation $\hat{e}$, and then obtain the binary verification label v by comparing $\hat{y}$ and y^*, i.e., $v = \mathbb{1}(\hat{y} = y^*)$. Next, after obtaining the set $\{(x, \hat{y}, \hat{e}, v)\}$, we augment it using the features discussed in Section 4.3.1 to construct $\{(x, \hat{y}, \hat{e}, \mathbb{F}(x), \mathbb{F}(\hat{y}), \mathbb{F}(\hat{e}), v)\}$, where $\mathbb{F}(*)$ denotes the set of features for each dimension. We can implement the verifier using any binary classification model. In our experiment, we evaluate the results using Support Vector Machine [13], Random Forest [32], and BERT [15] that assign a probability to the example as the verifier score.

Source of gold labels. One bottleneck to train the verifier model is to obtain gold labels. One can train high-quality classifiers with thousands or even more gold labels; however, this is entirely contrary to our goal of generating annotations cheaper and faster. To this end, we consider two different sources of gold labels: one is a small set (up to 500 labels) of gold labels from the same task, and the other is a relatively larger set of gold labels from another task in the same domain (e.g., PStance for SemEval where both are stance detection tasks). In our experiment, we report the results of using each one of these two sources and also those of using the two sources combined.

3.3 Step 3: Human Re-annotation

In this step, human annotators re-annotate the subset pinpointed by the verifier model in Step 2 (Figure 1 (right-bottom)). Basically, human annotators provide a label to the example for re-annotation. Previous works suggest that compared to fully manual annotation, providing LLM-generated labels to human annotators improves efficiency and quality of annotations [51, 61]. Also, AI-generated explanations may affect human annotator's behavior [19]. Hence, our objective is to see if demonstrating LLM prediction with or without accompanying explanations can further help human annotators understand the task and the prediction. Through a series of human-subject studies, we investigate the extent to which LLM-generated pre-labels and explanations facilitate human re-annotation practice. The results of these studies can inform us about the optimal re-annotation strategy for human annotators, which are presented in Section 5.

4 EXPERIMENTS: LLM ANNOTATION AND SUBSET SELECTION

We start by investigating our first research question: How do we assess the correctness of LLM-generated labels through the integration of signals from input data, LLM labels, and corresponding LLM explanations? To do so, given the input, we utilize LLM to label the data and generate explanations. Subsequently, we extract informative cues from these sources to construct a verifier model. A series of experiments are then conducted to confirm the effectiveness of our verifier models in assessing the accuracy of LLM-generated labels.

4.1 Datasets

In our experiment, we sample three general-purpose labeling (text classification) tasks where LLMs have demonstrated remarkable abilities but remain insufficient to entirely replace human annotation — natural language inference, stance detection, and hate speech detection. For each of these tasks, we select two representative datasets for evaluation. Finally, we randomly sample 2,000 data instances from each dataset while preserving the identical label distribution as found in the original dataset to ensure the representativeness of the selected data instances.

4.1.1 Natural language inference (NLI). NLI, also referred to as recognizing textual entailment (RTE), is the task of determining the inference relation between a hypothesis sentence and a premise sentence. This task serves as a unified framework for various NLP

Table 2: Prompt template including a task instruction, few-shot demonstrations with explanations, and the target input.

	Natural Language Inference	Stance Detection	Hate Speech Detection
Task instruction	Classify the inference relation between the premise sentence and the hypothesis sentence into entailment, contradiction, or neutral. Then explain your decision.	Classify the stance that the text holds towards the target into favor, against, or none. Then explain your decision.	Classify the post into hatespeech, offensive, or normal. Then explain your decision.
Demonstrations	Premise: Hypothesis: Answer: Explanation:	Text: Target: Answer: Explanation:	post: Answer: Explanation:
Target input	Premise: Hypothesis: Answer:	Text: Target: Answer:	post: Answer:
GPT's output	`<label>` Explanation: `<explanation>`	`<label>` Explanation: `<explanation>`	`<label>` Explanation: `<explanation>`

problems [57], necessitating human annotation to gather data for such a representational task.

Our evaluation centered on two well-known datasets: The Stanford Natural Language Inference (SNLI) corpus [8] and The Multi-Genre Natural Language Inference (MNLI) corpus [100]. SNLI is a collection of 570k human-written English sentence pairs manually annotated with the labels entailment, contradiction, and neutral. The corpus is curated to serve as a resource for developing and assessing text representation-learning systems. Modeled after the SNLI corpus, MNLI is another crowdsourced collection of 433k sentence pairs but differs in that it covers diverse spoken and written text genres such as fiction, letters, telephone speech, or 9/11 reports, which enables a cross-genre generalization evaluation.

4.1.2 Stance detection. The design of the stance detection task aims to determine from the text whether the author is in favor of the given target, against the given target, or whether neither is likely. This task facilitates diverse applications such as assessing consensus, interpersonal connections, social graphs, and public opinions in social contexts.

We evaluated stance detection on the earliest and most established SemEval-2016 Stance Dataset [62], which contains 2814 labeled tweets and their associated stance towards five targets: "Atheism", "Climate Change is a Real Concern", "Feminist Movement", "Hillary Clinton", and "Legalization of Abortion". The possible stance labels are favor, against, or none. Additionally, we utilized the P-Stance dataset [48], comprising 21,574 English tweets in the political domain, each annotated with a stance toward one of three targets: "Donald Trump", "Joe Biden", and "Bernie Sanders". Stances are categorized as favor or against.

4.1.3 Hate speech detection. Hate speech denotes language that disparages individuals or groups based on protected characteristics like race. Detecting and mitigating hate speech holds significant societal importance by curbing the dissemination of hateful ideas within networks.

Our evaluation included the Social Bias Inference Corpus (SBIC) [75] spanning over 34k implications about a thousand demographic groups. Hate speech is annotated with whether the social media posts were offensive or normal and with free-text explanations that highlight why a specific subgroup is targeted. We also incorporated the HateXplain [60] dataset. Each of around 20k posts in the dataset is annotated from three different perspectives: the 3-class classification (i.e., hatespeech, offensive, or normal), the target community, and the word and phrase level human rationales consisting of specific segments of the post that underpin their labeling decision.

4.2 LLM Annotation and Explanation

We use the Completion API[1] from OpenAI, configured with GPT module `text-davinci-003`, to annotate each dataset and to acquire explanations. Following configurations in previous work [44, 107], we formulate few-shot prompts for each dataset. To diversify explanations, we set the temperature parameter to 0.7, commonly used for creative tasks. Examples of the prompt template and GPT outputs are shown in Table 2.

4.3 Verifier for Subset Selection

As discussed in Section 3.2, a separate verifier model is used to score each annotation by the LLM and identify potential wrong labels, given the original example and labels and explanations from LLM as the input.

4.3.1 Feature selection. While one could regard the verification as another text classification task and train a verifier model with only the textual inputs, there are additional signals beyond the textual inputs that might be effective in locating potentially incorrect labels. Specifically, the LLM takes text samples as the input and generates labels and explanations as the output. This allows us to consider signals from three dimensions along the LLM annotation process: the data sample, the LLM-generated labels, and the LLM-generated explanations. In addition, prior work has shown that LLM performance can be affected by these dimensions, e.g. LLMs perform poorly on complex text [67], inputs similar to demonstration can improve LLMs' performance [54] (*sample* characteristics); LLMs predict specific classes better [107] (*label* characteristics); and Chain-of-Thought prompting improves LLMs' accuracy [97] (*explanations* characteristics). In our implementation, we extract

[1] https://platform.openai.com/docs/api-reference/completions

Table 3: Unified structure of prompts given to the human simulator proxy in x, $\hat{e}$, $x + \hat{e}$ settings, for the SNLI dataset. Black text is fixed prompts; blue denotes data content; purple denotes explanation content; teal denotes the human simulator's output.

	Input	Output
x	Question: What's the relationship between the premise sentence <premise> and the hypothesis sentence <hypothesis>? Choose A, B, or C. A: entailment B: neutral C: contradiction	<A/B/C>
$\hat{e}$	Question: What's the relationship between two sentences? Explanation: <explanation> Choose A, B, or C. A: entailment B: neutral C: contradiction	<A/B/C>
$x + \hat{e}$	Question: What's the relationship between the premise sentence <premise> and the hypothesis sentence <hypothesis>? Explanation: <explanation> Choose A, B, or C. A: entailment B: neutral C: contradiction	<A/B/C>

signals from these three dimensions and incorporate them into the training data of our verifier model.

- **Sample characteristics.** From a sample text, we extract text characteristics features and sample representative features. We utilize all textual features available from the TextDescriptives API [28], which offers descriptive statistics, syntactic complexity, and readability metrics. Further, we quantify coherence using the Coherence Momentum model [34]. Regarding sample representativeness, we assess how atypical the sample is compared to the others in the dataset by calculating an outlier score through the cleanlab.outlier API [39], and calculate semantic similarity to the demonstration examples in the prompt using BERTscore [105]. This gives us a total of 70 features about sample characteristics. We describe a subset of features as follows:
  - **Coherence:** Higher coherence refers to better logical connections in the text.
  - **Perplexity:** Higher perplexity of a sentence refers to more unexpected words from humans, indicating lower generation quality.
  - **Readability:** Higher readability indicates that it is harder for a human to understand the text.
- **Label characteristics.** The OpenAI Completion API provides logits for the top 5 predicted tokens at each output position. Thus, we leverage the corresponding logits to measure the uncertainty associated with GPT's label accuracy. We compute the logit of the label (the first output token), similar to Wang et al. [90], and the entropy for the top-5 logits. We also compute the average logit and average entropy across all tokens. This gives us a total of 7 features about label characteristics, including the GPT-predicted label.
- **Explanation characteristics.** For a generated explanation, we employ metrics based on uncertainty in generation, textual characteristics, and the quality of explanation. We first measure the uncertainty associated with GPT-generated explanations. This involves utilizing the logit values provided by the API to compute both the logit and entropy for the first token within the explanation, as well as the average logit and average entropy across all tokens within the explanation. Next, we extract textual features by following a similar procedure as applied to the data sample. Another vital metric for evaluating the quality of explanations is their faithfulness to the GPT-generated labels, i.e., explanation's helpfulness towards a model's decision. For example, an LLM-generated explanation that is not supportive of the predicted label can give a signal on the uncertainty of the label. Inspired by prior research [10, 29, 102], we adopt two definitions of faithfulness:
  - **Sufficiency:** An explanation is sufficient if it contains enough information to enable a human to predict the model output $\hat{y}$ solely from the explanation $\hat{e}$, expressed as $p(\hat{y}|\hat{e})$.
  - **Simulatability:** This metric measures the extent to which an explanation $\hat{e}$ conveys extra semantic content that informs the human about the task model's output $\hat{y}$ in the context of its input x. It quantifies the change in the prediction performance of a human simulator based on whether the explanation $\hat{e}$ is included with the input x, denoted as $\max(0, p(\hat{y}|x + \hat{e}) - p(\hat{y}|x))$. A high simulatability score indicates that when provided with both the explanation $\hat{e}$ and the input x, a human is more likely to accurately predict the task model's output $\hat{y}$, in contrast to when she has only the input x.

  These two metrics are implemented as follows. To simulate human judgment, we employ a pre-trained flan-T5 model as a proxy. To extract the confidence score of the $p(\hat{y}|\hat{e})$, $p(\hat{y}|x + \hat{e})$, $p(\hat{y}|x)$, as depicted in Table 3, we use different prompts in the three settings and instruct the flan-T5 model to provide responses restricted to certain choices. Treating the flan-T5 model's output as a multi-label classification, we derive the confidence values through softmax applied to logits associated with those choices. All the above approaches give us a total of 73 features about explanation characteristics.

4.3.2 Model training. After obtaining the features for each training example as discussed above, we standardize each of the 150 features to real numbers with a mean of 0 and a standard deviation of 1, such that they can be fed to classifiers. We implement verifier

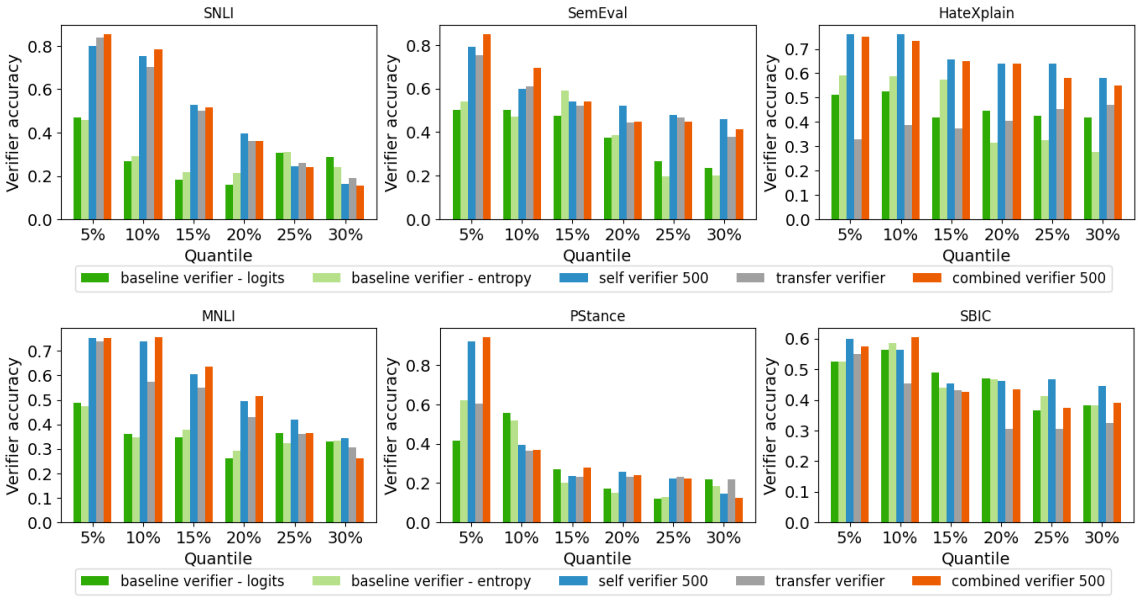


Figure 2: Verifier accuracy in bottom quantiles. The percentage in the x-axis denotes the quantile: e.g., 5% denotes the examples with the bottom (0.00, 0.05] verification score, 10% denotes the examples with the bottom (0.05, 0.10] verification score. SELF and COMBINED verifiers achieve similar performance and consistently outperform baseline verifiers, while TRANS does not work well for hate speech detection.

models using different model architectures for binary classification, including two traditional machine learning methods, SVM [13] and Random forest classifiers [32], from the Scikit-learn library [70], and one language model based classifier, based on the Huggingface's BERT [15] implementation [1]. The BERT-based classifier takes as input both the original text in the training example and the selected features. We add an MLP classifier on top of the BERT-encoder (which transforms the original text to 768-dimensional embeddings) with our extracted features as additional input. More implementation details are in the Appendix.

Among these trained classifier-based verifiers, the random forest verifier has clearly better overall accuracy. As we do not focus on tuning the verifier model, we only include the Random Forest verifier's result in this section to represent trained verifiers. Note that the BERT-encoder is also fine-tuned in this case, and it is common for random forests to outperform deep learning methods for such a relatively small dataset. The comparison between all trained verifiers can be found in the Appendix (Figure 1), including the case where the BERT-encoder is not fine-tuned (taking the embedding as a set of features, Figure 3 in the Appendix). Besides, we also evaluate baseline verifiers that simply use the logits and entropies from GPT's first token as the score.

As we discussed in Section 3.2.2, training the verifier would require a few gold labels, which would not be possible if we considered the annotation task from scratch. Therefore, we consider three settings that we believe are reasonable for using this collaborative annotation framework in practice.

SELF: The verifier is trained on the data from the same task of the same domain. To be more practical for a new annotation task and also to reduce the cost, we consider small sets of training data ranging from 100 examples to 500 examples.

TRANS: The verifier is trained on the data from the same task but a different dataset. This domain-adaptation style training is common when people want to build models when an existing dataset is available. We use 2,000 examples from the other datasets.

COMBINED: the verifier is trained on both the data from the same dataset and the data from another dataset, which combines the two above settings.

The baseline verifiers will not be trained and will not take any data as input. For each dataset we evaluate, we randomly sample 2,000 examples from the original training set with the same label distribution. Then, we sample 500 examples from those 2,000 for training and the rest for testing. We evaluate three different train-test splits and report the averaged results. For each dataset used for TRANS, we also randomly sample 2,000 examples from the original training set with the same label distribution. All results in Section 4.4 are averaged over 3 runs using different random seeds.

4.4 Results

Verifier accuracy. For our verifier models trained as classifiers, their classification results are nearly perfect in terms of recall, i.e., identify almost all the wrong GPT annotations. However, they struggle with poor precision and return false positives. Like the baseline verifiers that use single metrics, setting a fixed threshold to decide potentially wrong annotations is tricky and requires further human efforts. Since our goal is not to re-examine all LLM annotations and the re-annotation process usually depends on the

Table 4: Accuracy of verifiers for bottom 100, 200, or 300 instances. Our verifier outperforms all compared methods. BV-l denotes the baseline verifier using GPT's logit for the first token, and BV-e denotes the baseline verifier using GPT's entropy for the first token. Datasets are grouped by tasks. Grouped datasets are used for transfer training for each other.

	100					200					300				
	BV-l	BV-e	Self	Trans	Combined	BV-l	BV-e	Self	Trans	Combined	BV-l	BV-e	Self	Trans	Combined
SNLI	0.44	0.44	0.77	0.80	0.85	0.33	0.35	0.73	0.71	0.75	0.27	0.30	0.62	0.60	0.63
MNLI	0.47	0.48	0.76	0.70	0.76	0.41	0.41	0.72	0.62	0.73	0.36	0.37	0.65	0.57	0.66
SBIC	0.54	0.54	0.59	0.53	0.57	0.53	0.52	0.54	0.49	0.55	0.51	0.50	0.52	0.44	0.51
HateXplain	0.50	0.60	0.77	0.32	0.75	0.48	0.58	0.73	0.36	0.72	0.47	0.52	0.70	0.37	0.69
SemEval	0.50	0.51	0.75	0.72	0.82	0.49	0.53	0.67	0.64	0.73	0.46	0.50	0.61	0.58	0.63
PStance	0.46	0.63	0.80	0.56	0.80	0.43	0.49	0.56	0.43	0.57	0.35	0.37	0.45	0.36	0.46

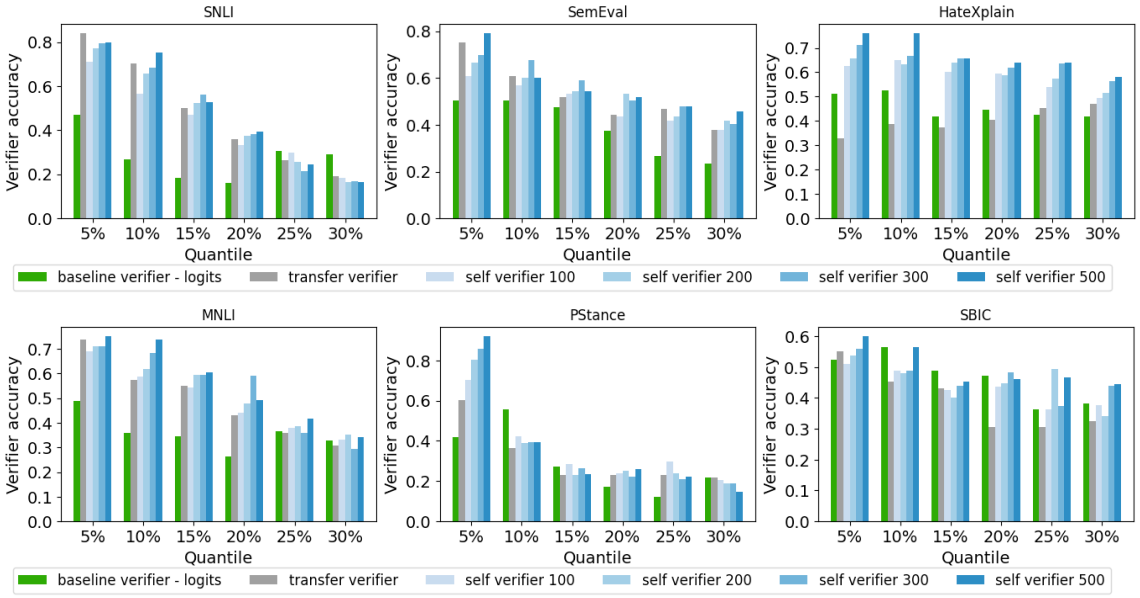


Figure 3: Self verifier accuracy varying number of gold labels. The percentage in the x-axis denotes the quantile: e.g., 5% denotes the examples with the bottom (0.00, 0.05] verification score, 10% denotes the examples with the bottom (0.05, 0.10] verification score. In most cases, Self can achieve clearly better performance than the baseline verifier with only 200 gold labels.

cost in practice, we take a deeper analysis by looking into the bottom examples ranked by each verifier — we expect those examples to be re-annotated by humans. Figure 2 plots the mean accuracy of each verifier for bottom quantiles (each contains 75 examples), demonstrating the effectiveness of our verifiers. In all datasets, our trained verifiers can identify wrong labels more accurately than the baseline verifiers, especially for the "worst" annotations in the bottom 5% or 10%. For NLI datasets (SNLI and MNLI), since these two datasets are quite similar, the performance of Trans is comparable with Self and Combined. For the hate speech detection task, Trans is not performing that well due to their dissimilarity. The performance of Combined indicates that using the existing dataset is still sometimes helpful (e.g., the performance gap between Self and Combined for stance detection in SemEval dataset is 6% and 8% for the bottom 5% and 10% examples, resp.). We also report the verifier accuracy for all bottom 100, 200, and 300 examples in Table 4.

Comparing different data sources. Figure 3 plots the accuracy of the random forest based Self verifier varying the number of gold labels from the same dataset. Although using all 500 gold labels shows the best overall performance as expected, with only 100 or 200 gold labels, the verifier model achieves comparable accuracy. For Combined verifiers, Figure 4 indicates the same trend for most datasets when increasing the number of gold labels from the same dataset, with 2,000 gold labels from the other dataset of the same task.

Re-annotation after verification. To demonstrate the effectiveness of having such a verifier in our annotation framework, we plot the overall accuracy against different numbers of re-annotation in Figure 5. For example, 10% on the x-axis in the figure indicates that the bottom 10% of examples ranked by the corresponding verifier

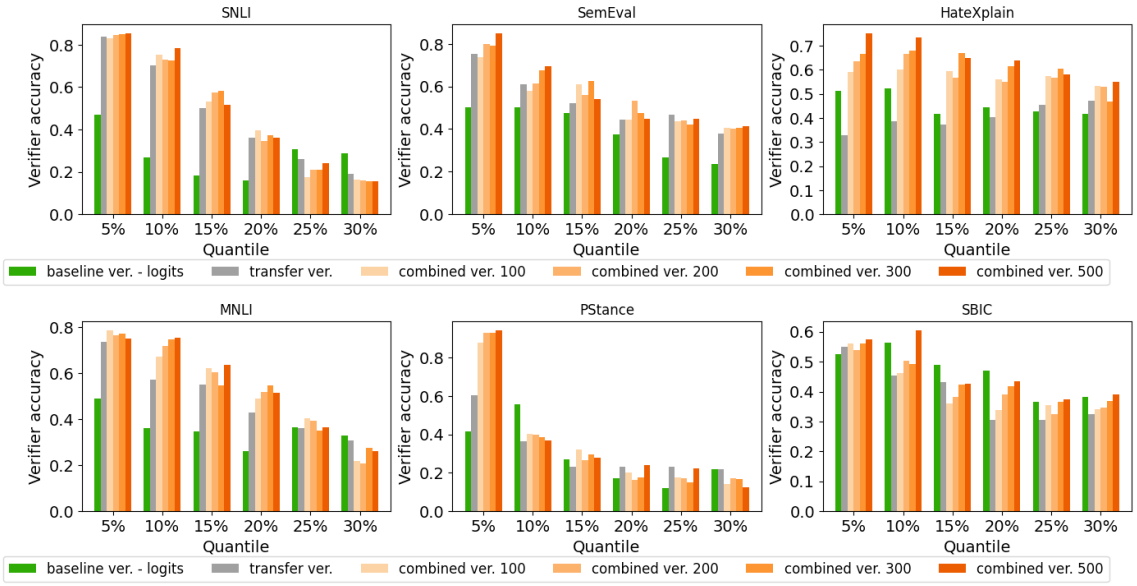


Figure 4: COMBINED verifier accuracy varying number of gold labels. The percentage in the x-axis denotes the quantile: e.g., 5% denotes the examples with the bottom (0.00, 0.05] verification score, 10% denotes the examples with the bottom (0.05, 0.10] verification score. With the other dataset, COMBINED can achieve overall good performance with only 100 gold labels from the current dataset.

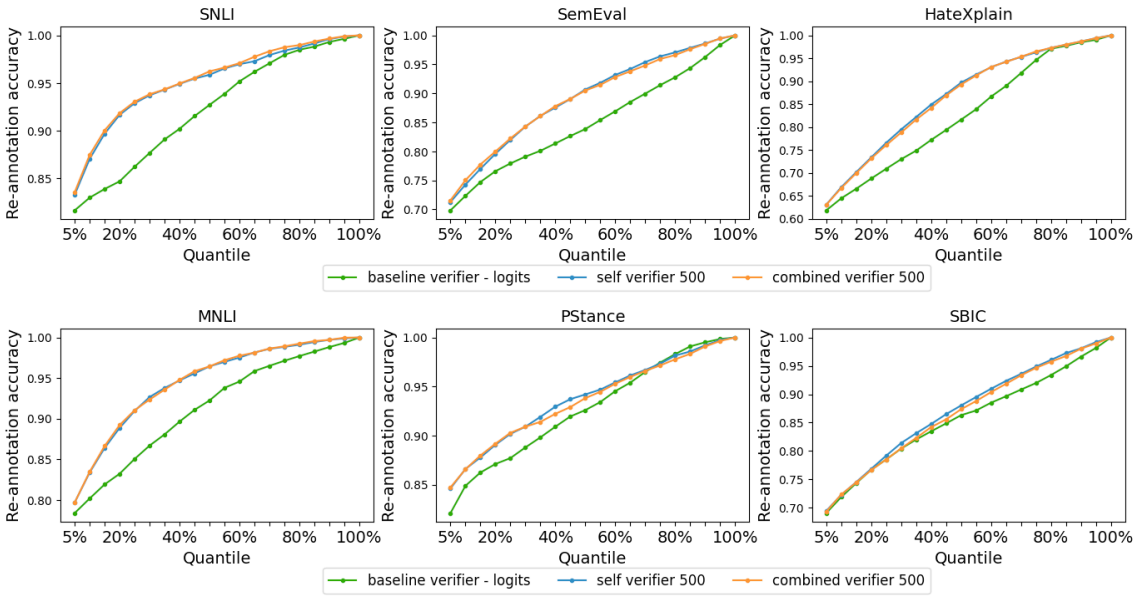


Figure 5: Overall accuracy after re-annotation vs. percentage of re-annotation.

are re-annotated. In this figure, we assume that there is an oracle re-annotation. Comparing our verifiers (orange line and blue line) with the overall GPT accuracy, it is clear that our verifiers can potentially improve the overall accuracy on all the test examples by at least 10% with only re-annotating 15% "bad" labels, which has a large margin compared with the baseline verifier (green line) in 4 out of 6 datasets.

Cost associated with training a verifier. For a fair comparison, we conduct a cost analysis for using verifiers. We denote the total annotation cost as the sum of *LLM inference cost*, *verifier cost*, and

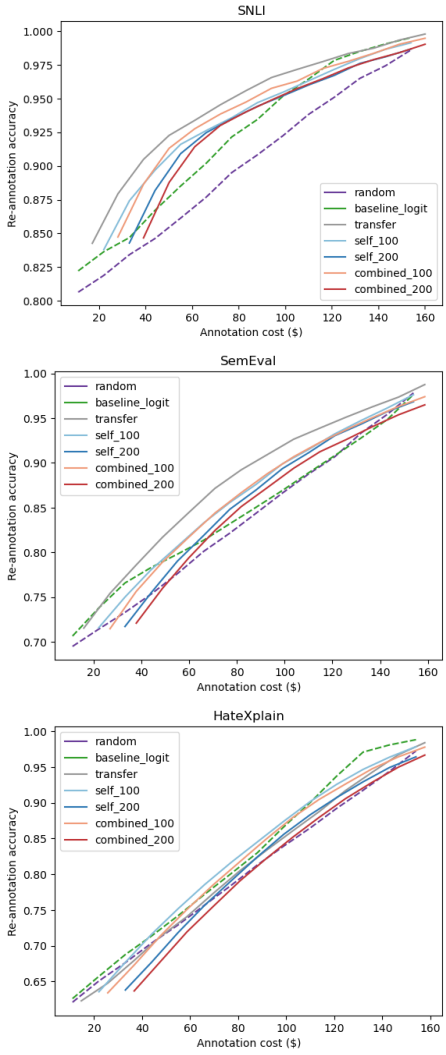


Figure 6: Accuracy of the entire dataset after re-annotation vs. total annotation cost. Our verifiers are shown as solid lines. Dashed lines represent baseline verifiers.

human re-annotation cost. Note that we only consider the non-computational cost listed above and exclude computation cost or miscellaneous annotator cost (e.g., recruiting and training humans) from our analysis. Also, in our framework, LLMs first annotate all data, i.e., the *LLM inference cost* will be the same regardless of the choice of a verifier, and thus will be ignored. Thus, we compare *verifier cost* + *human re-annotation cost* for each verifier. Following Wang et al. [90], we approximate the cost of human labeling as \$1.1/instance. GPT inference cost for a sample is calculated as the average number of tokens[2] in generated prompts * \$0.02/1,000 tokens. The costs are calculated as follows: *Verifier cost* = \$1.1 * the number of gold labels for training + (in case of Combined/Trans) *LLM inference cost for transfer dataset. Human re-annotation cost* = \$1.1 * the number of re-annotation.

Figure 6 shows *verifier cost* + *human re-annotation cost* associated with each verifier. For SNLI and HateXplain datasets, baseline verifiers work better under a low-budget setting, while with a modest budget (e.g. more than \$40-\$60), our verifiers clearly outperform the others. On the other hand, Trans consistently outperforms the baseline verifiers in the entire cost range for SNLI and SemEval datasets. An optimal strategy can be using the entropy-based verifier to select and annotate a small number of instances and then using the labeled subset to train Self or Combined. As the size of the unlabeled dataset is bigger, the more useful our verifier will be due to the high verifier accuracy (Table 4). One can also take advantage of Trans if it is available from previously annotated datasets.

5 HUMAN-SUBJECT STUDY: HUMAN RE-ANNOTATION STRATEGY

The objective of the verifier is to identify a subset of potentially erroneous LLM-generated labels, so as to avoid wasting human efforts on re-annotating correct LLM labels. Subsequently, this identified subset of LLM-generated labels is presented to human annotators for validation, who are tasked with either confirming their correctness or correcting them if necessary. In the performance evaluation presented in Section 4.4, we make an assumption that during the re-annotation phase, the re-annotators *always* provide accurate labels. However, in practical scenarios, human annotators are susceptible to making errors. Thus, a human-subject study involving real human annotators becomes necessary.

Another crucial aspect that needs consideration is what is the optimal strategy for enhancing human re-annotation performance: (1) not presenting any LLM model's outputs to human annotators, (2) presenting only the LLM-generated labels, and (3) presenting both the LLM-generated labels and the explanations. Answers to this question turn out to be nuanced. For example, humans might be influenced or misled by LLM's labels and explanations. Conversely, the LLM explanations may serve as a tool in revealing the LLM's erroneous reasoning processes to human annotators.

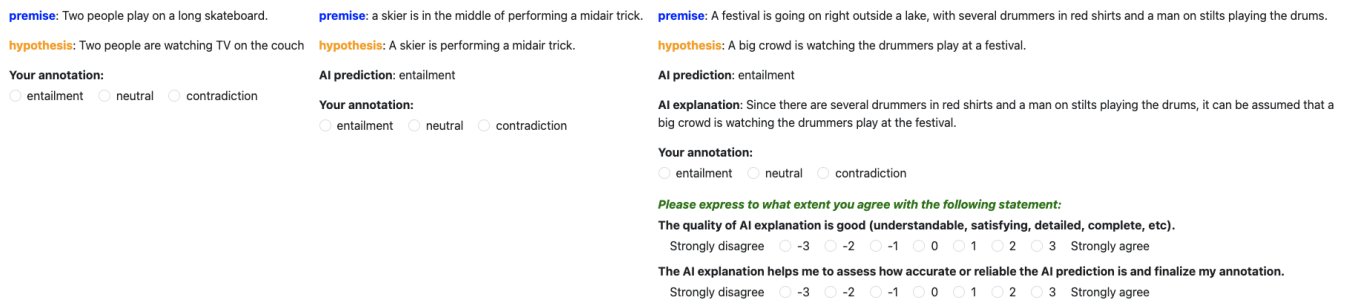
To comprehensively address these questions, we conducted an extensive online human-subject study. This study not only provides a holistic evaluation of our verifier's performance in real-world conditions but also affords insights into the most effective strategy to assist human annotators in their re-annotation tasks.

5.1 Study Design

5.1.1 Re-annotation tasks. In this human-subject study, we asked participants to complete a set of re-annotation tasks with or without help from LLM. Specifically, we selected two datasets out of the six datasets in Section 4.1: SNLI for the natural language inference task, and SemEval-2016 for the stance detection task.

5.1.2 Experimental treatments. Our study adopted a mixed design by having types of verifier (random, baseline, and our verifier) and score quantiles (bottom 1st-100th, 101st-200th, 201st-300th) as between-subject variables, and LLM assistance strategies (No LLM assistance, LLM label, LLM label & explanation) as a within-subject variable.

[2]We use the tiktoken tokenizer provided by OpenAI.



(a) No LLM assistance **(b) With LLM label** **(c) With LLM label & explanation**

Figure 7: Task interface used in the human-subject study for different re-annotation strategies.

Type of verifier / Score quantile. For each dataset, we selected the bottom 20% instances with the lowest scores from each type of verifier, i.e., random, baseline, and our verifier. For the selected 300 instances from each verifier, we further divide them into three quantiles (6.67% of each dataset) based on verifier scores, i.e., bottom 1st-100th, 101st-200th, 201st-300th. This resulted in nine conditions – each combination of verifier type and score quantile, denoted by ['random100', 'random200', 'random300', 'baseline100', 'baseline200', 'baseline300', 'ours100', 'ours200', 'ours300']. When LLM labels are shown, participants may observe LLM accuracy varies across conditions. For instance, the majority of LLM labels in 'ours100' (e.g., the bottom 100 instances with lowest scores from our verifier) could be incorrect, while a much higher proportion of LLM labels selected by the random verifier may be correct. Thus, to account for the potential impact of observed AI accuracy on participants' performance [103], for these conditions, we adopted a between-subject design so that each participant was assigned instances from only one type of verifier and only one range of score quantile for the assigned verifier.

LLM assistance. We further adopted a within-subject design for the re-annotation strategies, and each participant would experience all three re-annotation strategies. For each annotation task and each verifier, we created 3 treatments by varying whether and how the LLM assists human annotators with the re-annotation:

- **No LLM assistance:** Participants are not presented with any output from the LLM on each data instance.
- **LLM label:** Participants are presented with the label generated by the LLM on each data instance.
- **LLM label & explanation:** Participants are presented with both the label and the explanation generated by the LLM on each data instance.

We opted for a within-subject design because of findings in prior research [7] suggesting that repeated exposure to the same AI assistance style may lead users to focus on case features rather than a specific AI assistance, diminishing the effects of AI assistance style. Exposure to multiple AI assistance styles is also realistic in real-world contexts where AI explanations are presented selectively [43] and progressively [80] based on user needs.

Participants were randomly assigned to types of verifier and verifier score quantiles. For each participant, the order of LLM assistance treatments was also randomized. Per LLM assistance treatment, each instance was annotated by three human annotators, while each human annotator annotated 10 instances. This resulted in 30 unique participants in each of the nine between-subject conditions. The 30 data instances each participant saw were different across three LLM assistance treatments. The order of LLM assistance treatments, as well as the order of data instances within each treatment, were randomized across participants.

5.1.3 Experimental procedure. Participants first completed a demographic survey. Next, participants were presented with an interactive tutorial designed to guide them through their assigned annotation task, e.g., natural language inference or stance detection.[3] Within this tutorial, we asked a series of qualification questions to ensure participants had a clear understanding of the task instructions. After completing the tutorial, participants were presented with the study background as follows: an AI model is used to annotate data, which can be incorrect, and a human annotator should re-annotate potentially incorrect instances. Participants then proceeded to annotate a set of 30 data instances, divided into 3 phases with 10 instances in each phase. Each phase corresponded to one of the three within-subject treatments: no LLM assistance, LLM label, or LLM label & explanation. For each data instance, participants followed a two-step procedure. Initially, they reviewed the data instance, with or without the LLM prediction and explanation, and then provided a final annotation, as illustrated in Figure 7. Notably, the participants did not receive any accuracy feedback concerning their annotations or the LLM's annotations on any of these data instances. Before submitting their study responses, participants were required to complete an exit survey. In the exit survey, participants were asked to report their perceived cognitive load, satisfaction with the annotation experience, and self-assessed accuracy for each phase. In addition, for the two phases with LLM assistance, participants were asked to rate their intention to utilize the LLM assistance again, their perceived helpfulness of the LLM, their perceived accuracy of the LLM, and their trust in the LLM (see the Appendix for more detail). All ratings were on 7-point Likert Scale. To ensure response quality, the study also included three attention-check questions. Participants were instructed to select pre-specified options in these questions, which later helped us to filter out the data from inattentive participants.[4]

[3]We used the task instructions in the original datasets [8, 62].
[4]Screenshots of our study interface are provided in the Appendix.

Table 5: Accuracy of labels after re-annotating bottom 100, 200, or 300 instances from each verifier. Our verifier outperforms all compared methods.

#anno	0	100			200			300		
Verifier	LLM	Random	Baseline	Ours	Random	Baseline	Ours	Random	Baseline	Ours
SNLI	0.795	0.805	0.811	**0.831**	0.811	0.819	**0.865**	0.814	0.824	**0.868**
SemEval	0.663	0.665	0.680	**0.699**	0.672	0.692	**0.729**	0.681	0.707	**0.741**

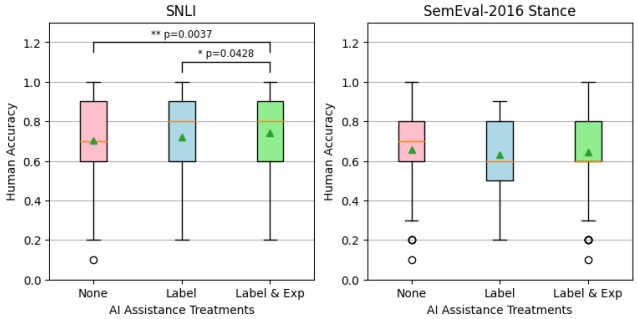


Figure 8: Accuracy of individual annotators by AI assistance treatments. Green markers indicate average accuracy.

5.1.4 Participants. We launched two studies in Prolific[5], one for natural language inference task (SNLI dataset) and one for stance detection task (SemEval2016 dataset). We selected participants who are located in the United States and have a past approval rate ≥ 98% and ≥ 1,000 approved submissions. For each study, we recruited participants until the number of participants who passed attention checks reached 270 (=30 participants * 9 between-subject conditions). Each participant was allowed to participate only once. The percentages of participants who passed attention checks are 78.95% for natural language inference task and 88.82% for stance detection task.[6] Participants were paid $1.6USD for participation and up to $0.9USD as a performance-based bonus.[7] To motivate participants to pay attention during re-annotation, we paid $0.03 USD for each correct annotation to participants with ≥ 70% overall accuracy.

5.2 Results

5.2.1 Verifier evaluation. For end-to-end evaluation of our framework, we reported the accuracy of collected labels after human re-annotation in Table 5. Each instance selected by a verifier was re-annotated by three crowd workers per AI assistance treatment. Results were based on aggregated labels via majority voting. Our verifier outperformed all compared methods by a big margin. For the SNLI dataset, LLM's accuracy was 79.5%, after re-annotation, the final accuracy was 86.8% with a 7.3% improvement. For the SemEval2016 dataset, LLM's accuracy was 66.3%, after re-annotation, the final accuracy was 74.1% with a 7.8% improvement.

[5]https://www.prolific.co/
[6]Details about demographic characteristics of study participants are provided in the Appendix.
[7]The pay is calculated based on pilot studies that took approximately 12 minutes to complete.

5.2.2 Effect of LLM assistance. We investigated our second research question: Do LLM-generated labels and explanations help humans in re-annotation? Figure 8 shows the average accuracy of participants for each AI assistance treatment. For the SNLI dataset, human accuracy was higher when providing both LLM labels and explanations (74.15%) than when only providing LLM labels (72.07%) or without any assistance (70.44%). On the other hand, results on the stance detection task did not show any statistically significant differences between AI assistance treatments. We further analyzed whether the effect of LLM assistance differed on LLM correct instances and LLM wrong instances. We compared average human accuracy in two groups of instances: one in which the LLM label was accurate and another where it was incorrect. For the instances where LLM was correct, participants were more accurate with more LLM assistance (Figure 9). In contrast, when LLM was incorrect, providing the wrong LLM labels hurted human accuracy. There were no statistically significant differences between showing the LLM explanation when the LLM label was wrong.

We analyzed exit survey ratings on participants' perceived cognitive load, satisfaction with the annotation experience, self-assessed accuracy, intention to use AI assistance again, perceived AI helpfulness, perceived AI accuracy, and their trust in AI (Figure 10). We performed ANOVA test for each category and a follow-up pairwise analysis between AI assistance treatments. For both tasks, participants agreed that having both LLM labels and explanations is more mentally demanding than the alternatives ($p < .001$), likely because they had to read each explanation and assess its validity. For the SNLI dataset, the self-assessed accuracy of participants was higher for LLM-assisted treatments (T1<T2,T3) and they thought having both labels and explanations was more helpful than having only labels. Conversely, for the SemEval2016 Stance dataset, participants believed that their accuracy was higher when only labels were given than both labels and explanations were provided. They were also less satisfied when both labels and explanations were presented, compared with the alternatives. Upon further examination, we found that the quality of explanations generated for the SemEval dataset was lower, as illustrated in Figure 11. Clearly, there were more explanations in the SemEval dataset with lower coherence, and it was even lower when the GPT annotation was wrong. While in the SNLI dataset, the distribution of coherence was close for correct and incorrect annotations. Interestingly, these two datasets showed different distributions for sufficiency and readability. For the SemEval dataset, the explanations were less sufficient and more readable when the GPT annotation was wrong. When the GPT annotation was wrong, the explanations in the SemEval dataset had lower perplexity than the correct annotation, contrasting the distribution of the SNLI dataset. Finally, there were no statistically

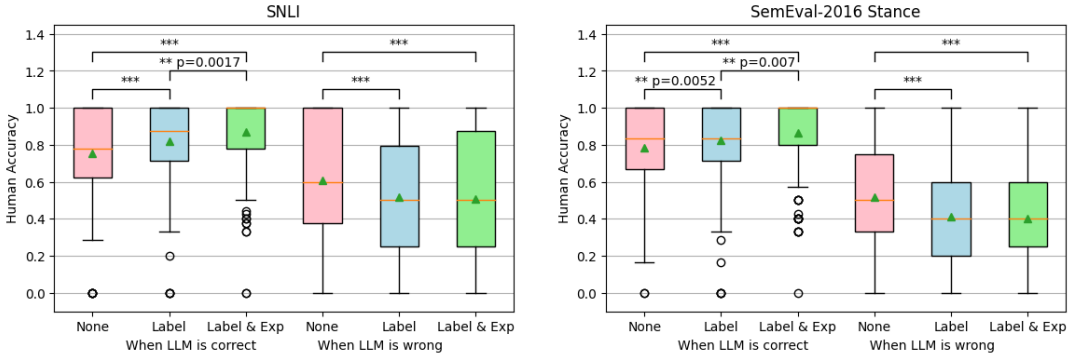
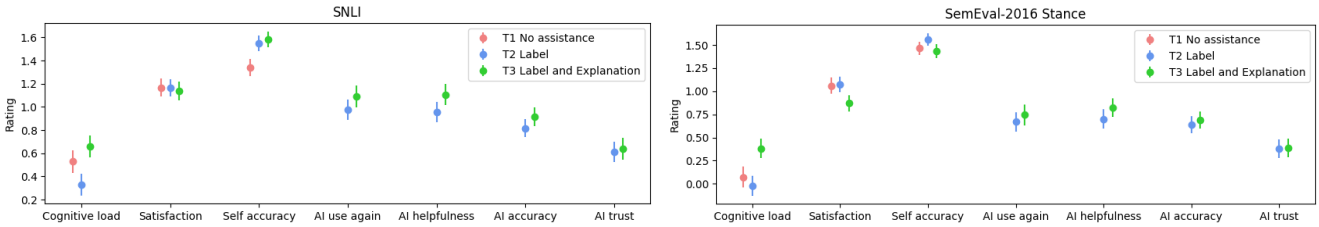


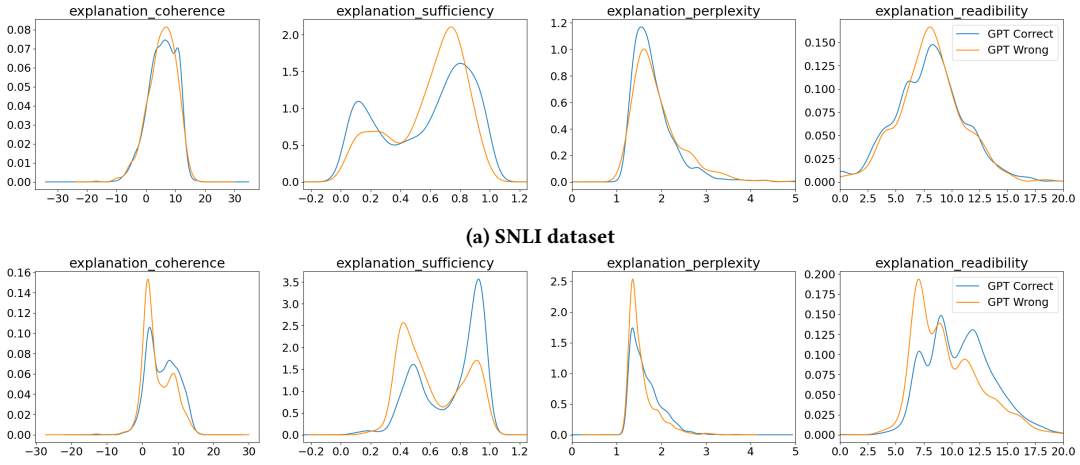
Figure 9: Accuracy of individual annotators on instances where LLM is correct (left three boxes) or wrong (right three boxes) by AI assistance treatments. Green markers indicate average accuracy. * indicates $p < 0.001$.**



(a) Natural language inference task

(b) Stance detection task

Figure 10: Exit survey ratings by AI assistance treatments.



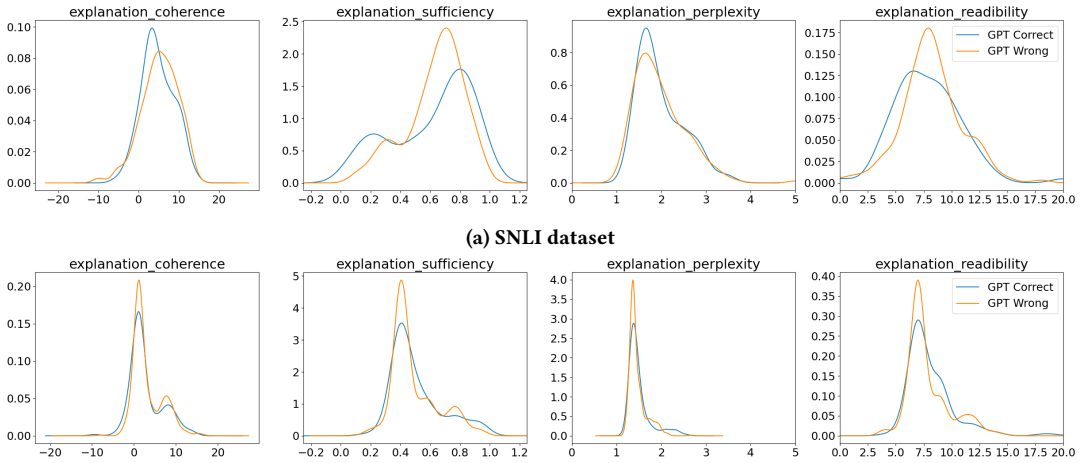
(a) SNLI dataset

(b) SemEval dataset

Figure 11: Distribution (density function) of explanation quality measures grouped by the correctness of GPT annotation. For perplexity and readability, lower is better; for the rest metrics, higher is better.

significant effects of AI assistance treatments on participants' intention to use AI again, perceived accuracy of the assistant AI model, and their trust in the AI model for both tasks.

5.2.3 Effect of types of verifiers and score quantiles. We studied whether human re-annotation accuracy differed by the types of verifiers (random, baseline, and ours) and score quantiles (0-100, 101-200, and 201-300), and whether the effect of LLM assistance interacted with these conditions. Figure 13 illustrate that human performance was lowest on instances selected by our verifier and second lowest on the baseline verifier. This was likely due to our verifier identifying more incorrect LLM labels, meaning harder



(a) SNLI dataset

(b) SemEval dataset

Figure 12: Distribution (density function) of explanation quality measures grouped by the correctness of GPT annotation for bottom 200 examples selected by our verifier. For perplexity and readability, lower is better; for the rest metrics, higher is better.

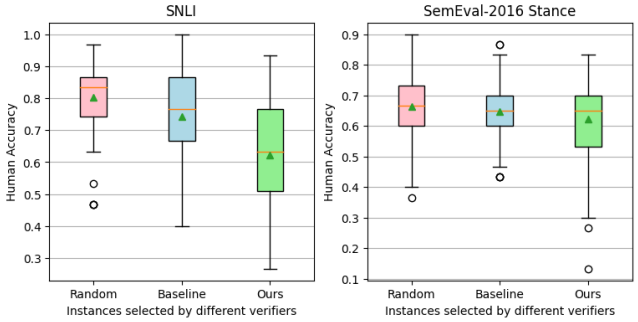


Figure 13: Accuracy on instances identified by each verifier. Green markers indicate average accuracy.

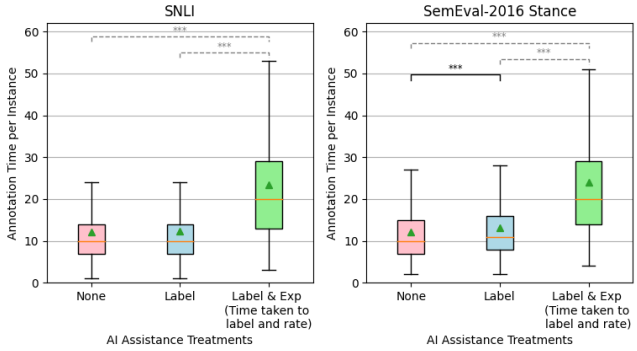


Figure 14: Annotation time per task instance by AI assistance treatments. Annotation time for label & explanation treatment includes time taken for both labeling and rating explanations, and hence is not suitable for direct comparison (grayed out). Green markers indicate average accuracy. * indicates $p < 0.001$.**

instances for LLMs were assigned to the annotators of this group. There was no statistically significant difference between different verifiers and/or score quantiles on the effect of LLM assistance treatments.

5.2.4 Effect on annotation time. We further examined whether annotation time was affected by different factors. We calculated the annotation time for a task as the time taken between opening up the task page in our web interface and clicking a button to move to the next page.[8] Our study setting did not guarantee that the participants were actively engaging in annotation throughout the study duration. Hence, annotation tasks with long response times, which might indicate the participants were idle, were excluded from analysis. We also ignored annotation times exceeding 1.5 interquartile range (IQR) below the lower quartile or above the upper quartile of a participant [72]. The remaining tasks were 7,538 for SNLI and 7,461 for SemEval out of 8,100.

We conducted Mann-Whitney U test for pair-wise analysis between AI assistance treatments. Figure 14 shows the average annotation time taken to label one task instance for each AI assistance treatment. We found that annotation time was slower when LLM labels were shown (13.20 seconds) than without any assistance (12.06 seconds, 8.7% faster). In contrast, there was no statistically significant difference between the two treatments for the SNLI dataset. For both datasets, annotating with both LLM labels and explanations shown was statistically significantly slower than the other treatments, but we could not presume whether it was because of additional cognitive load to read explanation or time taken to give ratings to the explanations. Further human-subject study is

[8]Unfortunately, our interface did not capture precise annotation time for the label & explanation treatment. This was because annotators need to label and rate the explanation quality in the same task page as in Figure 7c. We still reported the annotation times in the figures but excluded them in analysis.

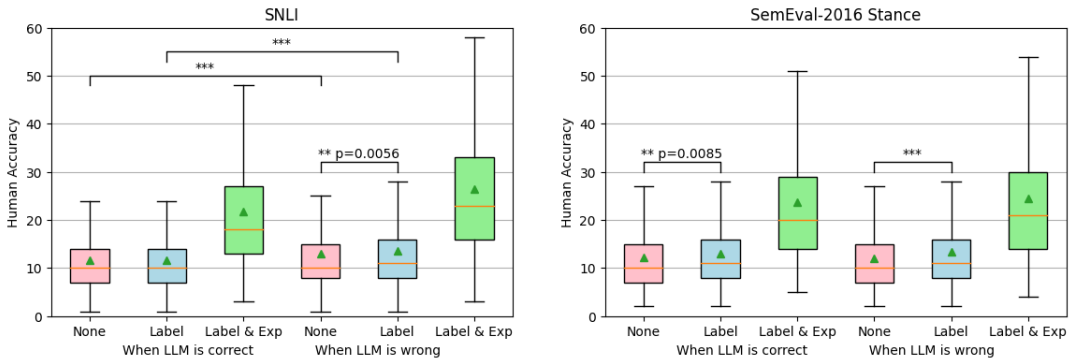


Figure 15: Annotation time per task instance where LLM is correct (left three boxes) or wrong (right three boxes) by AI assistance treatments. Annotation time for label & explanation treatment includes time taken for both labeling and rating explanations, and hence is not suitable for direct comparison. Green markers indicate average accuracy. * indicates $p < 0.001$.**

required to analyze annotation time difference between the label & explanation treatment from the others. Additionally, we examined if the effect of AI assistance treatment varied by the correctness of LLM labels. For both datasets, when LLM labels were incorrect, annotation time was statistically significantly slower when LLM labels were shown than without any assistance (Figure 15). This was unsurprising since incorrect AI pre-labels could make human annotators reconsider their decisions, leading to additional time spent. For the SNLI dataset, humans were slower when LLM was wrong regardless of AI assistance. We posit that instances where LLM was wrong were also difficult to humans.

6 DISCUSSION

6.1 Verifying LLM Labels

Logits are not enough. Our empirical evaluation clearly shows that our trained verifiers can largely outperform the logits-based baseline verifiers in most cases. This is expected because LLMs tend to be over-confident [101], and the token-level log probabilities may not align with the label space precisely. The textual features of both input samples and explanations provide much richer information than the logits. Furthermore, as LLM-generated reasoning steps or explanations may reveal errors or inconsistencies in its prediction [47], verifiers can learn more implicit signals of the LLM generation.

Ways to improve verifiers. We conducted an ablation study by adding different sets of features starting with label characteristics (detailed results in the Appendix, Figure 2). The findings reveal a clear improvement in verifier performance when adding more sets of features, especially for explanation characteristics. This confirms our design choice of incorporating three dimensions of features in the input and underscores the effectiveness of leveraging LLM-generated explanation.

Our verifier may be further improved by adding even more features. We have explored the effectiveness of augmenting our verifier with text embeddings generated from pre-trained language models like BERT. The results demonstrate that utilizing text embeddings as additional features can often lead to improved verifier accuracy (see the Appendix, Figure 3), indicating that there is still space for improving the verifier. Additionally, we posit that certain tasks may benefit from task-specific features. For instance, we can add factuality metrics for knowledge-intensive tasks [68], demographic bias metrics for bias detection [14], or toxicity metrics for hate speech detection tasks using any language models pre-trained with existing knowledge [22]. If the auxiliary language models are substantially smaller than LLMs, the computation cost to obtain those additional features is negligible compared to sampling each LLM annotation.

Alternative approaches to verify LLM labels. We also test alternative methods to identify potentially incorrect LLMs without training a verifier model. First, we can directly ask LLMs to score their confidence, i.e., self-assessment. However, we notice that they generate seemingly plausible but not faithful scores, which is a common phenomenon called LLM hallucination. Another technique is instructing LLMs to output a 'Do not know' label when they are not confident about their predictions. Next, we can prompt LLM multiple times for the same input and measure annotation agreement, i.e., self-consistency [11]. For these approaches, we cannot control the number of instances to re-annotate. Also, for commercial LLMs, the cost of LLM annotation will be multiplied by the number of trials for the self-consistency approach, making those methods requiring multiple trials undesirable.

6.2 Human Re-annotation with LLM Assistance

Annotators rely on LLM assistance. When LLM labels are incorrect, manually annotating is more accurate than having any kind of assistance in almost all combinations of conditions, e.g., for annotators who got a highly accurate subset of LLM labels (>80%) as well as ones who got mostly inaccurate LLM labels (<30%). This indicates participants can be easily misled by LLM assistance, regardless of LLM accuracy. As shown in Figure 12, the distributions of explanation sufficiency diverge less for correct and incorrect predictions for the bottom 200 examples compared to the distributions for the entire dataset (Figure 11). Further study is required to understand which factor can mitigate this.

Re-annotation and explanation quality. Our crowdsourced study results in Section 5.2.2 show that for both datasets, when

the LLM is wrong, human accuracy was significantly lower when providing both LLM labels and explanations than when providing only the example. This may be due to the ambiguity of those wrong annotations: the majority of them are 'neutral'/'none'. Figure 12 shows that the sufficiency of explanations are lower when the LLM is wrong, potentially indicating lower explanation quality (SemEval also has clearly lower coherence). While low-quality explanations can be a good indicator of incorrect LLM annotation, they may be less helpful in assisting human annotation. Similarly, the SemEval dataset even shows no difference across the three AI assistance treatments. By comparing the metrics across two datasets in Figure 12, although SNLI explanations are more complex (higher perplexity and worse readability), they have better semantic qualities (coherence and sufficiency). We conjecture this is the reason that the crowd workers have better performance with LLM explanations for the SNLI dataset and are more satisfied with the explanations. Improving the explanation quality, especially for tasks with domain-specific knowledge, could be another essential problem in human-LLM collaboration.

Presentation of LLM Explanations. Our study finds that having LLM explanations is strongly mentally demanding (Figure 10). In our setup, we utilize natural language explanations, which often is longer than the input text. Consequently, reading an explanation and labeling takes more time than manual labeling, which results in a high cognitive load. Compared to natural language explanations, extractive explanations [58] such as highlighted text, can reduce cognitive load while conveying some justification of LLM labels to the annotators. Future work can investigate the effect of different types and presentations of LLM explanations in human re-annotation.

Towards efficient human re-annotation of LLM labels. An ideal annotation system should be accurate, cost-effective, and generalizable. In pursuit of these goals, we explored strategies to leverage the extensive knowledge embedded in LLMs and maximize the impact of human annotation efforts. In our study, the *verifier* model serves the role of identifying potentially incorrect LLM labels. The selection of a verifier should be guided by the allocated budget, balancing between the cost of training an accurate yet expensive verifier and the waste of human re-annotation on correctly labeled samples.

Furthermore, to improve the quality of human re-annotations and elicit cleaner labels, we can start with non-LLM-assisted human re-annotation on a small number of data instances. Ideally, instances with the lowest scores from the verifier are more likely to be truly incorrect. According to our user study results, when LLM labels are incorrect, human annotations without any LLM assistance are the most accurate. Furthermore, the *verifier* itself can, in turn, be enhanced by having a feedback loop based on the human re-annotations to allow for more relevant and diverse data points to be 'verified'. The current design of Lapras focuses on the annotation step for training downstream models, while we may improve the verifier in an iterative process, with feedback from the human annotators and signals from the progressively trained downstream models. In addition, there are possibilities to reduce the cost of human efforts further. For example, we can instruct LLMs to self-verify or re-annotate their own labels in another round before the human re-annotation process.

6.3 Design Implication for Human-LLM Collaborative Annotation Tools

To apply our Human-LLM collaborative annotation framework to annotation tools, there are a few practical considerations to keep in mind.

Improving the quality of LLM labels and explanations. We chose prompt templates used in previous works to obtain LLM annotations and explanations. In practice, LLM performance for a task may significantly vary based on the choice of prompt templates. Therefore, it is essential for human-LLM collaborative annotation tools to have prompt tuning capabilities. In addition to improving the accuracy of LLM labels, prompt engineering techniques to improve the self-rationalization performance (i.e., faithful explanations) should be considered.

Robust post-processing of LLM outputs. Despite explicit instructions in the prompt and even utilizing few-shot examples, the LLM predictions can sometimes be random. For example, the predicted label may not match with any of the label options provided for the task (eg. predicting '*neutral*' when the instruction stated to label the sentiment of a task as either '*positive*' or '*negative*' only. Furthermore, users may instruct an LLM to produce a label and explanation in a certain format e.g. JSON. However, the output produced by LLMs may not always adhere to this specific format. Hence, it becomes necessary to build and design a post-processing module to retrieve the LLM annotations and ensure they fit the requirements of the task and prompt instructions.

Re-annotation interface with selective LLM assistance. Depending on the characteristics of the given task and dataset to annotate, the effect of LLM assistance may differ. Re-annotation interface should be flexible enough to change when and what kind of information to show to annotators. For example, by default, we may not provide any assistance, and offer assistance only when we have explanations of good quality.

Human annotator selection. In subjective tasks, the human re-annotation process would need a form of majority voting among different annotators. Furthermore, in tasks that require specific domain knowledge, it becomes necessary to include crowd workers who are experts in that task domain, as otherwise, it would hamper the re-annotation process and verification model.

6.4 Limitations

There are a few limitations in this work. First, as we conducted the crowdsourced study on Prolific, our findings may not generalize to other platforms and annotation by human experts. Second, the accuracy of the verifier depends on the gold labels for training in our setting, and the verifier may adapt poorly to other datasets, while we can indeed dynamically update the verifier as the human annotation comes in. Third, as discussed in Section 6.3, investigating better prompts for annotation and explanation generation could improve both LLM accuracy and the quality of generated explanations, even leading to better training signals for the verifiers. Another limitation is the prevailing bias that exists in the data the LLMs are trained on, such as racial, cultural, and gender bias. Such biases may seep into the label predictions and explanations made by the LLM, which raises several ethical concerns. Furthermore, the datasets we work on may deal with sensitive issues, such as hatespeech, which may be

sensitive to some users. Lastly, for commercial LLMs, data privacy and other legal considerations need to be kept in mind.

7 CONCLUSION AND FUTURE WORK

As large language models (LLMs) have been widely used for data annotation, it is important to understand its defects and improve the annotation quality through collaboration with human annotators. In this paper, we studied how to design LLM-human collaborative annotation frameworks by leveraging LLM's label and self-explanation in verification and re-annotation. We built our Lapras system and conducted empirical experiments and human-subject studies to evaluate the verification and re-annotation steps in Lapras. Our findings indicate the effectiveness of our verifier with LLM-generated explanations. Further, to enhance the human-LLM collaboration, crowd workers' annotation performance and feedback tell us that there is a need to quantify and improve the quality of LLM explanations and carefully decide when explanations are helpful for human re-annotation.

ACKNOWLEDGMENTS

We would like to thank Dan Zhang, Hayate Iso, Pouya Pezeshkpour, Estevam Hruschka, Eser Kandogan, Tom Mitchell, and our colleagues in Megagon Labs for their valuable comments on this work. We are grateful to all anonymous reviewers who provided many helpful comments.

REFERENCES

[1] [n. d.]. BERT base model (uncased). https://huggingface.co/bert-base-uncased. Accessed: 2023-09-14.

[2] [n. d.]. Introducing ChatGPT. https://openai.com/blog/chatgpt. Accessed: 2023-09-14.

[3] Mykhaylo Andriluka, Jasper R. R. Uijlings, and Vittorio Ferrari. 2018. Fluid Annotation: A Human-Machine Collaboration Interface for Full Image Annotation. In *Proceedings of the 26th ACM International Conference on Multimedia* (Seoul, Republic of Korea) *(MM '18)*. Association for Computing Machinery, New York, NY, USA, 1957–1966. https://doi.org/10.1145/3240508.3241916

[4] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. PaLM 2 Technical Report. https://doi.org/10.48550/ARXIV.2305.10403

[5] Gagan Bansal, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel Weld. 2021. Does the Whole Exceed its Parts? The Effect of AI Explanations on Complementary Team Performance. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 81, 16 pages. https://doi.org/10.1145/3411764.3445717

[6] Meghana Moorthy Bhat, Alessandro Sordoni, and Subhabrata Mukherjee. 2021. Self-training with Few-shot Rationalization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 10702–10712. https://doi.org/10.18653/v1/2021.emnlp-main.836

[7] Reuben Binns, Max Van Kleek, Michael Veale, Ulrik Lyngs, Jun Zhao, and Nigel Shadbolt. 2018. 'It's Reducing a Human Being to a Percentage': Perceptions of Justice in Algorithmic Decisions. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3173574.3173951

[8] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 632–642. https://doi.org/10.18653/v1/D15-1075

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[10] Samuel Carton, Anirudh Rathore, and Chenhao Tan. 2020. Evaluating and Characterizing Human Rationales. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 9294–9307. https://doi.org/10.18653/v1/2020.emnlp-main.747

[11] Angelica Chen, Jason Phang, Alicia Parrish, Vishakh Padmakumar, Chen Zhao, Samuel R. Bowman, and Kyunghyun Cho. 2024. Two Failures of Self-Consistency in the Multi-Step Reasoning of LLMs. *Transactions on Machine Learning Research* (2024). https://openreview.net/forum?id=5nBqY1y96B

[12] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. https://doi.org/10.48550/ARXIV.2110.14168

[13] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning* 20, 3 (Sept. 1995), 273–297. https://doi.org/10.1007/bf00994018

[14] Paula Czarnowska, Yogarshi Vyas, and Kashif Shah. 2021. Quantifying Social Biases in NLP: A Generalization and Empirical Comparison of Extrinsic Fairness Metrics. *Transactions of the Association for Computational Linguistics* 9 (2021), 1249–1267. https://doi.org/10.1162/tacl_a_00425

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[16] Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, and Lidong Bing. 2023. Is GPT-3 a Good Data Annotator?. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Toronto, Canada, 11173–11195. https://doi.org/10.18653/v1/2023.acl-long.626

[17] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A Survey on In-context Learning. https://doi.org/10.48550/ARXIV.2301.00234

[18] Mingming Fan, Xianyou Yang, TszTung Yu, Q. Vera Liao, and Jian Zhao. 2022. Human-AI Collaboration for UX Evaluation: Effects of Explanation and Synchronization. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW1, Article 96 (apr 2022), 32 pages. https://doi.org/10.1145/3512943

[19] Sharon A Ferguson, Paula Akemi Aoyagui, and Anastasia Kuzminykh. 2023. Something Borrowed: Exploring the Influence of AI-Generated Explanation Text on the Composition of Human Explanations. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI EA '23)*. Association for Computing Machinery, New York, NY, USA, Article 253, 7 pages. https://doi.org/10.1145/3544549.3585727

[20] Karën Fort and Benoît Sagot. 2010. Influence of Pre-Annotation on POS-Tagged Corpus Development. In *Proceedings of the Fourth Linguistic Annotation Workshop*. Association for Computational Linguistics, Uppsala, Sweden, 56–63. https://aclanthology.org/W10-1807

[21] Simret Araya Gebreegziabher, Zheng Zhang, Xiaohang Tang, Yihao Meng, Elena L. Glassman, and Toby Jia-Jun Li. 2023. PaTAT: Human-AI Collaborative Qualitative Coding with Explainable Interactive Rule Synthesis. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 362, 19 pages. https://doi.org/10.1145/3544548.3581352

[22] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 3356–3369. https://doi.org/10.18653/v1/2020.findings-emnlp.301

[23] Reza Ghaeini, Sadid A. Hasan, Vivek Datla, Joey Liu, Kathy Lee, Ashequl Qadir, Yuan Ling, Aaditya Prakash, Xiaoli Fern, and Oladimeji Farri. 2018. DR-BiLSTM: Dependent Reading Bidirectional LSTM for Natural Language Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 1460–1469. https://doi.org/10.18653/v1/N18-1132

[24] Tiash Ghosh, Ratul Kishore Saha, Mamata Jenamani, Aurobinda Routray, Sanjai Kumar Singh, and Arpita Mondal. 2023. SeisLabel: An AI-Assisted Annotation Tool for Seismic Data Labeling. In *IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium*. 5081–5084. https://doi.org/10.1109/IGARSS52108.2023.10283015

[25] 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 (2023), e2305016120. https://doi.org/10.1073/pnas.2305016120

[26] Ben Green and Yiling Chen. 2019. Disparate Interactions: An Algorithm-in-the-Loop Analysis of Fairness in Risk Assessments. In *Proceedings of the Conference*

on Fairness, Accountability, and Transparency (Atlanta, GA, USA) *(FAT* '19)*. Association for Computing Machinery, New York, NY, USA, 90–99. https://doi.org/10.1145/3287560.3287563

[27] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*. PMLR, 1321–1330. https://proceedings.mlr.press/v70/guo17a.html

[28] Lasse Hansen, Ludvig Renbo Olsen, and Kenneth Enevoldsen. 2023. TextDescriptives: A Python package for calculating a large variety of metrics from text. *Journal of Open Source Software* 8, 84 (2023), 5153. https://doi.org/10.21105/joss.05153

[29] Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. 2020. Leakage-Adjusted Simulatability: Can Models Generate Non-Trivial Explanations of Their Behavior in Natural Language?. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 4351–4367. https://doi.org/10.18653/v1/2020.findings-emnlp.390

[30] Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2023. AnnoLLM: Making Large Language Models to Be Better Crowdsourced Annotators. https://doi.org/10.48550/ARXIV.2303.16854

[31] Andres Hernandez, Harry Hochheiser, John Horn, Rebecca Crowley, and Richard Boyce. 2014. Testing Pre-Annotation to Help Non-Experts Identify Drug-Drug Interactions Mentioned in Drug Product Labeling. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* 2, 1 (Oct. 2014), 14–15. https://doi.org/10.1609/hcomp.v2i1.13213

[32] Tin Kam Ho. 1995. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, Vol. 1. 278–282 vol.1. https://doi.org/10.1109/ICDAR.1995.598994

[33] Sahil Jayaram and Emily Allaway. 2021. Human Rationales as Attribution Priors for Explainable Stance Detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 5540–5554. https://doi.org/10.18653/v1/2021.emnlp-main.450

[34] Prathyusha Jwalapuram, Shafiq Joty, and Xiang Lin. 2022. Rethinking Self-Supervision Objectives for Generalizable Coherence Modeling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 6044–6059. https://doi.org/10.18653/v1/2022.acl-long.418

[35] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language Models (Mostly) Know What They Know. https://doi.org/10.48550/ARXIV.2207.05221

[36] Sunnie S. Y. Kim, Elizabeth Anne Watkins, Olga Russakovsky, Ruth Fong, and Andrés Monroy-Hernández. 2023. "Help Me Help the AI": Understanding How Explainability Can Support Human-AI Interaction. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 250, 17 pages. https://doi.org/10.1145/3544548.3581001

[37] Ronald T. Kneusel and Michael C. Mozer. 2017. Improving Human-Machine Cooperative Visual Search With Soft Highlighting. *ACM Trans. Appl. Percept.* 15, 1, Article 3 (sep 2017), 21 pages. https://doi.org/10.1145/3129669

[38] Ziyi Kou, Lanyu Shang, Yang Zhang, Zhenrui Yue, Huimin Zeng, and Dong Wang. 2022. Crowd, Expert & AI: A Human-AI Interactive Approach Towards Natural Language Explanation Based COVID-19 Misinformation Detection. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. International Joint Conferences on Artificial Intelligence Organization, 5087–5093. https://doi.org/10.24963/ijcai.2022/706 AI for Good.

[39] Johnson Kuan and Jonas Mueller. 2022. Back to the Basics: Revisiting Out-of-Distribution Detection Baselines. https://doi.org/10.48550/ARXIV.2207.03061

[40] Tsung-Ting Kuo, Jina Huh, Jihoon Kim, Robert El-Kareh, Siddharth Singh, Stephanie Feudjio Feupe, Vincent Kuri, Gordon Lin, Michele E. Day, Lucila Ohno-Machado, and Chun-Nan Hsu. 2018. The Impact of Automatic Pre-annotation in Clinical Note Data Element Extraction - the CLEAN Tool. https://doi.org/10.48550/ARXIV.1808.03806

[41] Taja Kuzman, Igor Mozetič, and Nikola Ljubešić. 2023. ChatGPT: Beginning of an End of Manual Linguistic Data Annotation? Use Case of Automatic Genre Identification. https://doi.org/10.48550/ARXIV.2303.03953

[42] Vivian Lai and Chenhao Tan. 2019. On Human Predictions with Explanations and Predictions of Machine Learning Models: A Case Study on Deception Detection. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (Atlanta, GA, USA) *(FAT* '19)*. Association for Computing Machinery, New York, NY, USA, 29–38. https://doi.org/10.1145/3287560.3287590

[43] Vivian Lai, Yiming Zhang, Chacha Chen, Q. Vera Liao, and Chenhao Tan. 2023. Selective Explanations: Leveraging Human Input to Align Explainable AI. *Proc. ACM Hum.-Comput. Interact.* 7, CSCW2, Article 357 (oct 2023), 35 pages. https://doi.org/10.1145/3610206

[44] Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. 2022. Can language models learn from explanations in context?. In *Findings of the Association for Computational Linguistics: EMNLP 2022*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 537–563. https://doi.org/10.18653/v1/2022.findings-emnlp.38

[45] Dong-Ho Lee, Rahul Khanna, Bill Yuchen Lin, Seyeon Lee, Qinyuan Ye, Elizabeth Boschee, Leonardo Neves, and Xiang Ren. 2020. LEAN-LIFE: A Label-Efficient Annotation Framework Towards Learning from Explanation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Online, 372–379. https://doi.org/10.18653/v1/2020.acl-demos.42

[46] Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, Wenhu Chen, and Xifeng Yan. 2022. Explanations from Large Language Models Make Small Reasoners Better. https://doi.org/10.48550/ARXIV.2210.06726

[47] Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023. Making Language Models Better Reasoners with Step-Aware Verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Toronto, Canada, 5315–5333. https://doi.org/10.18653/v1/2023.acl-long.291

[48] Yingjie Li, Tiberiu Sosea, Aditya Sawant, Ajith Jayaraman Nair, Diana Inkpen, and Cornelia Caragea. 2021. P-Stance: A Large Dataset for Stance Detection in Political Domain. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Online, 2355–2365. https://doi.org/10.18653/v1/2021.findings-acl.208

[49] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Alexander Cosgrove, Christopher D Manning, Christopher Re, Diana Acosta-Navas, Drew Arad Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue WANG, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Andrew Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic Evaluation of Language Models. *Transactions on Machine Learning Research* (2023). https://openreview.net/forum?id=iO4LZibEqW Featured Certification, Expert Certification.

[50] Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching Models to Express Their Uncertainty in Words. *Transactions on Machine Learning Research* (2022). https://openreview.net/forum?id=8s8K2UZGTZ

[51] Todd Lingren, Louise Deleger, Katalin Molnar, Haijun Zhai, Jareen Meinzen-Derr, Megan Kaiser, Laura Stoutenborough, Qi Li, and Imre Solti. 2013. Evaluating the impact of pre-annotation on annotation speed and potential bias: natural language processing gold standard development for clinical named entity recognition in clinical trial announcements. *Journal of the American Medical Informatics Association* 21, 3 (09 2013), 406–413. https://doi.org/10.1136/amiajnl-2013-001837

[52] Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. WANLI: Worker and AI Collaboration for Natural Language Inference Dataset Creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 6826–6847. https://doi.org/10.18653/v1/2022.findings-emnlp.508

[53] Han Liu, Vivian Lai, and Chenhao Tan. 2021. Understanding the Effect of Out-of-distribution Examples and Interactive Explanations on Human-AI Decision Making. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2, Article 408 (oct 2021), 45 pages. https://doi.org/10.1145/3479552

[54] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What Makes Good In-Context Examples for GPT-3?. In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Association for Computational Linguistics, Dublin, Ireland and Online, 100–114. https://doi.org/10.18653/v1/2022.deelio-1.10

[55] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Comput. Surv.* 55, 9, Article 195 (jan 2023), 35 pages. https://doi.org/10.1145/3560815

[56] Shuai Ma, Ying Lei, Xinru Wang, Chengbo Zheng, Chuhan Shi, Ming Yin, and Xiaojuan Ma. 2023. Who Should I Trust: AI or Myself? Leveraging Human and AI Correctness Likelihood to Promote Appropriate Trust in AI-Assisted Decision-Making. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 759, 19 pages. https://doi.org/10.1145/3544548.3581058

[57] Bill MacCartney and Christopher D. Manning. 2008. Modeling Semantic Containment and Exclusion in Natural Language Inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. Coling 2008 Organizing Committee, Manchester, UK, 521–528. https://aclanthology.org/C08-1066

[58] Bodhisattwa Prasad Majumder, Oana Camburu, Thomas Lukasiewicz, and Julian Mcauley. 2022. Knowledge-Grounded Self-Rationalization via Extractive and Natural Language Explanations. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*. PMLR, 14786–14801. https://proceedings.mlr.press/v162/majumder22a.html

[59] Ana Marasovic, Iz Beltagy, Doug Downey, and Matthew Peters. 2022. Few-Shot Self-Rationalization with Natural Language Prompts. In *Findings of the Association for Computational Linguistics: NAACL 2022*. Association for Computational Linguistics, Seattle, United States, 410–424. https://doi.org/10.18653/v1/2022.findings-naacl.31

[60] Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 17 (May 2021), 14867–14875. https://doi.org/10.1609/aaai.v35i17.17745

[61] Marie Mikulová, Milan Straka, Jan Štěpánek, Barbora Štěpánková, and Jan Hajic. 2022. Quality and Efficiency of Manual Annotation: Pre-annotation Bias. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 2909–2918. https://aclanthology.org/2022.lrec-1.312

[62] Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. SemEval-2016 Task 6: Detecting Stance in Tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, 31–41. https://doi.org/10.18653/v1/S16-1003

[63] Nikita Nangia and Samuel R. Bowman. 2019. Human vs. Muppet: A Conservative Estimate of Human Performance on the GLUE Benchmark. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 4566–4575. https://doi.org/10.18653/v1/P19-1449

[64] Ansong Ni, Srini Iyer, Dragomir Radev, Veselin Stoyanov, Wen-Tau Yih, Sida Wang, and Xi Victoria Lin. 2023. LEVER: Learning to Verify Language-to-Code Generation with Execution. In *Proceedings of the 40th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 202)*. PMLR, 26106–26128. https://proceedings.mlr.press/v202/ni23b.html

[65] Philip Ogren, Guergana Savova, and Christopher Chute. 2008. Constructing Evaluation Corpora for Automated Clinical Named Entity Recognition. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. European Language Resources Association (ELRA), Marrakech, Morocco. http://www.lrec-conf.org/proceedings/lrec2008/pdf/796_paper.pdf

[66] OpenAI. 2023. GPT-4 Technical Report. https://doi.org/10.48550/ARXIV.2303.08774

[67] Simon Ott, Konstantin Hebenstreit, Valentin Liévin, Christoffer Egeberg Hother, Milad Moradi, Maximilian Mayrhauser, Robert Praas, Ole Winther, and Matthias Samwald. 2023. ThoughtSource: A central hub for large language model reasoning data. *Scientific Data* 10, 1 (Aug. 2023). https://doi.org/10.1038/s41597-023-02433-3

[68] Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding Factuality in Abstractive Summarization with FRANK: A Benchmark for Factuality Metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 4812–4829. https://doi.org/10.18653/v1/2021.naacl-main.383

[69] Nicholas Pangakis, Samuel Wolken, and Neil Fasching. 2023. Automated Annotation with Generative AI Requires Validation. https://doi.org/10.48550/ARXIV.2306.00176

[70] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, 85 (2011), 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html

[71] Eric Ringger, Marc Carmen, Robbie Haertel, Kevin Seppi, Deryle Lonsdale, Peter McClanahan, James Carroll, and Noel Ellison. 2008. Assessing the Costs of Machine-Assisted Corpus Annotation through a User Study. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. European Language Resources Association (ELRA), Marrakech, Morocco. http://www.lrec-conf.org/proceedings/lrec2008/pdf/832_paper.pdf

[72] Peter J. Rousseeuw and Mia Hubert. 2011. Robust statistics for outlier detection. *WIREs Data Mining and Knowledge Discovery* 1, 1 (2011), 73–79. https://doi.org/10.1002/widm.2

[73] Swarnadeep Saha, Peter Hase, Nazneen Rajani, and Mohit Bansal. 2022. Are Hard Examples also Harder to Explain? A Study with Human and Model-Generated Explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2121–2131. https://doi.org/10.18653/v1/2022.emnlp-main.137

[74] Gaurav Sahu, Pau Rodriguez, Issam Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data Augmentation for Intent Classification with Off-the-shelf Large Language Models. In *Proceedings of the 4th Workshop on NLP for Conversational AI*. Association for Computational Linguistics, Dublin, Ireland, 47–57. https://doi.org/10.18653/v1/2022.nlp4convai-1.5

[75] Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. 2020. Social Bias Frames: Reasoning about Social and Power Implications of Language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 5477–5490. https://doi.org/10.18653/v1/2020.acl-main.486

[76] Jianhao Shen, Yichun Yin, Lin Li, Lifeng Shang, Xin Jiang, Ming Zhang, and Qun Liu. 2021. Generate & Rank: A Multi-task Framework for Math Word Problems. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 2269–2279. https://doi.org/10.18653/v1/2021.findings-emnlp.195

[77] Maria Skeppstedt. 2013. Annotating named entities in clinical text by combining pre-annotation and active learning. In *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*. Association for Computational Linguistics, Sofia, Bulgaria, 74–80. https://aclanthology.org/P13-3011

[78] Maria Skeppstedt, Carita Paradis, and Andreas Kerren. 2016. PAL, a tool for Pre-annotation and Active Learning. *Journal for Language Technology and Computational Linguistics* 31, 1 (Jul. 2016), 81–100. https://doi.org/10.21248/jlcl.31.2016.203

[79] Brett R. South, Danielle Mowery, Ying Suo, Jianwei Leng, Óscar Ferrández, Stephane M. Meystre, and Wendy W. Chapman. 2014. Evaluating the effects of machine pre-annotation and an interactive annotation interface on manual de-identification of clinical text. *Journal of Biomedical Informatics* 50 (2014), 162–172. https://doi.org/10.1016/j.jbi.2014.05.002 Special Issue on Informatics Methods in Medical Privacy.

[80] Aaron Springer and Steve Whittaker. 2019. Progressive disclosure: empirically motivated approaches to designing effective transparency. In *Proceedings of the 24th International Conference on Intelligent User Interfaces* (Marina del Ray, California) *(IUI '19)*. Association for Computing Machinery, New York, NY, USA, 107–120. https://doi.org/10.1145/3301275.3302322

[81] Mallory C. Stites, Megan Nyre-Yu, Blake Moss, Charles Smutz, and Michael R. Smith. 2021. Sage Advice? The Impacts of Explanations for Machine Learning Models on Human Decision-Making in Spam Detection. In *Artificial Intelligence in HCI*. Springer International Publishing, Cham, 269–284.

[82] Sarkar Sujoy, Amrith Krishna, and Pawan Goyal. 2023. Pre-annotation Based Approach for Development of a Sanskrit Named Entity Recognition Dataset. In *Proceedings of the Computational Sanskrit & Digital Humanities: Selected papers presented at the 18th World Sanskrit Conference*. Association for Computational Linguistics, Canberra, Australia (Online mode), 59–70. https://aclanthology.org/2023.wsc-csdh.4

[83] Petter Törnberg. 2023. ChatGPT-4 Outperforms Experts and Crowd Workers in Annotating Political Twitter Messages with Zero-Shot Learning. https://doi.org/10.48550/ARXIV.2304.06588

[84] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. https://doi.org/10.48550/ARXIV.2302.13971

[85] Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. In *Thirty-seventh Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=bzs4uPLXvi

[86] Donghan Wang, Madalina Fiterau, Artur Dubrawski, Marilyn Hravnak, Gilles Clermont, and Michael Pinsky. 2014. Interpretable active learning in support of clinical data annotation. *Critical Care Medicine* 42, 12 (December 2014), 1552.

[87] Han Wang, Ming Shan Hee, Md Rabiul Awal, Kenny Tsu Wei Choo, and Roy Ka-Wei Lee. 2023. Evaluating GPT-3 Generated Explanations for Hateful Content Moderation. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*. International Joint Conferences on Artificial Intelligence Organization, 6255–6263. https://doi.org/10.24963/ijcai.2023/694 AI for Good.

[88] PeiFeng Wang, Aaron Chan, Filip Ilievski, Muhao Chen, and Xiang Ren. 2023. PINTO: Faithful Language Reasoning Using Prompt-Generated Rationales. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=WBXbRs63oVu

[89] Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. 2023. SCOTT: Self-Consistent Chain-of-Thought Distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Toronto, Canada,

5546–5558. https://doi.org/10.18653/v1/2023.acl-long.304

[90] Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Want To Reduce Labeling Cost? GPT-3 Can Help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 4195–4205. https://doi.org/10.18653/v1/2021.findings-emnlp.354

[91] Xinru Wang, Chen Liang, and Ming Yin. 2023. The Effects of AI Biases and Explanations on Human Decision Fairness: A Case Study of Bidding in Rental Housing Markets. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*. International Joint Conferences on Artificial Intelligence Organization, 3076–3084. https://doi.org/10.24963/ijcai.2023/343 Main Track.

[92] Xinru Wang, Zhuoran Lu, and Ming Yin. 2022. Will You Accept the AI Recommendation? Predicting Human Behavior in AI-Assisted Decision Making. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 1697–1708. https://doi.org/10.1145/3485447.3512240

[93] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=1PL1NIMMrw

[94] Xinru Wang and Ming Yin. 2021. Are Explanations Helpful? A Comparative Study of the Effects of Explanations in AI-Assisted Decision-Making. In *26th International Conference on Intelligent User Interfaces* (College Station, TX, USA) *(IUI '21)*. Association for Computing Machinery, New York, NY, USA, 318–328. https://doi.org/10.1145/3397481.3450650

[95] Xinru Wang and Ming Yin. 2022. Effects of Explanations in AI-Assisted Decision Making: Principles and Comparisons. *ACM Trans. Interact. Intell. Syst.* 12, 4, Article 27 (nov 2022), 36 pages. https://doi.org/10.1145/3519266

[96] Xinru Wang and Ming Yin. 2023. Watch Out for Updates: Understanding the Effects of Model Explanation Updates in AI-Assisted Decision Making. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 758, 19 pages. https://doi.org/10.1145/3544548.3581366

[97] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, Vol. 35. Curran Associates, Inc., 24824–24837. https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf

[98] Sarah Wiegreffe, Jack Hessel, Swabha Swayamdipta, Mark Riedl, and Yejin Choi. 2022. Reframing Human-AI Collaboration for Generating Free-Text Explanations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 632–658. https://doi.org/10.18653/v1/2022.naacl-main.47

[99] Sarah Wiegreffe, Ana Marasović, and Noah A. Smith. 2021. Measuring Association Between Labels and Free-Text Rationales. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 10266–10284. https://doi.org/10.18653/v1/2021.emnlp-main.804

[100] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 1112–1122. https://doi.org/10.18653/v1/N18-1101

[101] Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can LLMs Express Their Uncertainty? An Empirical Evaluation of Confidence Elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=gjeQKFxFpZ

[102] Bingsheng Yao, Prithviraj Sen, Lucian Popa, James Hendler, and Dakuo Wang. 2023. Are Human Explanations Always Helpful? Towards Objective Evaluation of Human Natural Language Explanations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Toronto, Canada, 14698–14713. https://doi.org/10.18653/v1/2023.acl-long.821

[103] Ming Yin, Jennifer Wortman Vaughan, and Hanna Wallach. 2019. Understanding the Effect of Accuracy on Trust in Machine Learning Models. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300509

[104] Ann Yuan, Daphne Ippolito, Vitaly Nikolaev, Chris Callison-Burch, Andy Coenen, and Sebastian Gehrmann. 2021. SynthBio: A Case Study in Faster Curation of Text Datasets. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. https://openreview.net/forum?id=Fkpr2RYDvI1

[105] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SkeHuCVFDr

[106] Yunfeng Zhang, Q. Vera Liao, and Rachel K. E. Bellamy. 2020. Effect of confidence and explanation on accuracy and trust calibration in AI-assisted decision making. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (Barcelona, Spain) *(FAT* '20)*. Association for Computing Machinery, New York, NY, USA, 295–305. https://doi.org/10.1145/3351095.3372852

[107] Yiming Zhu, Peixian Zhang, Ehsan-Ul Haq, Pan Hui, and Gareth Tyson. 2023. Can ChatGPT Reproduce Human-Generated Labels? A Study of Social Computing Tasks. https://doi.org/10.48550/ARXIV.2304.10145

[108] Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2024. Can Large Language Models Transform Computational Social Science? *Computational Linguistics* (02 2024), 1–55. https://doi.org/10.1162/coli_a_00502