

Let Guidelines Guide You: A Prescriptive Guideline-Centered Data Annotation Methodology

Federico Ruggeri¹, Eleonora Misino², Arianna Muti³, Katerina Korre⁴

Paolo Torroni¹, Alberto Barrón-Cedeño⁵

¹DISI, University of Bologna, ²Musixmatch, ³Bocconi University

⁴Archimedes/Athena RC, ⁵DIT, University of Bologna

¹{federico.ruggeri6, p.torroni}@unibo.it

²eleonora.misino@gmail.com, ³arianna.muti@unibocconi.it

⁴k.korre@athenarc.gr, ⁵a.barron@unibo.it

Abstract

We introduce Guideline-Centered Annotation Methodology (GCAM), a novel methodology designed to report the annotation guidelines associated with each data instance. GCAM addresses four key limitations of the standard application of the prescriptive annotation methodology by reducing the information loss during annotation, ensuring adherence to guidelines, and enabling the efficient reuse of annotated data across multiple tasks that rely on the same guidelines. We evaluate GCAM with a focus on text classification tasks through (i) a human annotation study and (ii) an experimental evaluation with several machine learning models. Our results highlight the advantages of GCAM from multiple perspectives, guaranteeing a transparent evaluation of the successful application of the prescriptive paradigm and enabling a fine-grained model error analysis.

Warning: This paper contains examples of offensive and explicit content.

1 Introduction

The widespread application of machine learning models and, in particular, Large Language Models (LLMs) has led to a surge in demand for data availability. This also encouraged research on data quality, annotation, and regulations. A notable example is the introduction of Data-Centric AI terminology (Zha et al., 2025) to encompass a suite of best practices for dataset development. This is due to relevant factors, such as data quantity reaching a plateau (Villalobos et al., 2024) and guaranteeing data quality for large-scale datasets is challenging (e.g., the abattoir-effect (Raji et al., 2020)). Several contributions underscore the importance of clear and well-structured guidelines

for producing high-quality data annotation, including annotators eligibility (Zhang et al., 2023), disagreement resolution (Mostafazadeh Davani et al., 2022; Sandri et al., 2023), human evaluation (Krishna et al., 2023), checklists for data maintainability and reproducibility (Geburu et al., 2021; Pushkarna et al., 2022; Díaz et al., 2022), recommendations for mitigating harmful biases or information leakage (Klie et al., 2024), and data annotation methodologies (Paullada et al., 2021; Peng et al., 2021; Santy et al., 2023). Researchers have recently framed data annotation methodologies into distinct annotation paradigms. Paradigms provide recommendations on how to structure the annotation process to address annotation issues, a well-known challenge in subjective tasks such as hate speech detection (Plank et al., 2014; Sap et al., 2022) and subjectivity detection (Riloff and Wiebe, 2003; Savinova and Moscoso Del Prado, 2023). Each paradigm is designed to support the specific intended use of the collected data.

Rottger et al. (2022) were the first to propose the prescriptive and the descriptive paradigms. The prescriptive annotation paradigm enforces a shared belief of the task conveyed into a unique set of annotation guidelines on which annotators are instructed. In contrast, the descriptive paradigm favors annotators' perspectives and beliefs, which can be reflected in the resulting annotation. Another notable annotation paradigm is the perspectivist one (Cabitza et al., 2023), where multiple ground truths for the same data instance are allowed, while sharing a unique set of annotation guidelines similar to those from the prescriptive paradigm.

The existing annotation methodologies following these paradigms are centered on text classification tasks where data instances are annotated with class labels (Sap et al., 2022; AlDayel and Magdy, 2021; Kenyon-Dean et al., 2018; Antici et al., 2024). We denote this approach as the Stan-

Term	Symbol	Definition
Data instance	x	Textual instance to be annotated.
Guideline	g	Textual definition describing an annotation criterion.
Guideline set (or guidelines)	$\mathcal{G}$	A collection of annotation criteria.
Guideline subset	$\mathcal{G}_x$	Guideline set associated with instance x , $\mathcal{G}_x \subseteq \mathcal{G}$.
Class set	$\mathcal{C}$	Set of classes associated with a classification task.
Class subset	$\mathcal{C}_x$	Class set associated with instance x , $\mathcal{C}_x \subseteq \mathcal{C}$.
SAM annotation function	f	Function mapping a data instance x to a class subset $\mathcal{C}_x$.
GCAM annotation function	h	Function mapping a data instance x to a guideline subset $\mathcal{G}_x$.
GCAM class grounding function	r	Function mapping a guideline $g \in \mathcal{G}_x$ to a class $c \in \mathcal{C}$.

Table 1: Notation of the main concepts discussed in this article.

standard Annotation Methodology (SAM). When considering an annotation paradigm that relies on annotation guidelines, like the prescriptive one, we argue that SAM has four main limitations that can notably affect the quality of the data: (i) an *information loss* during data annotation, where the specific components of the guidelines used by annotators are not explicitly recorded; (ii) the *lack of transparency* in evaluating annotator adherence to the guidelines, making it difficult to assess whether annotation decisions were based on the provided instructions or subjective interpretations; (iii) the *task-specific annotations* resulting from SAM, which ties annotations to a task-specific class set, limiting the reuse of annotated data for other tasks that share the same annotation guidelines, unless undergoing additional human annotation efforts; and (iv) *unclear model adherence to guidelines*, where machine learning (ML) models are trained with class supervision only, despite using data annotated according to specific guidelines, thus making unclear to what extent model learning aligns with the task description conveyed by the guidelines. We elaborate on our hypothesis by analyzing the prescriptive annotation paradigm in text classification tasks, where convergence to the shared belief is an indicator of data quality and annotation guidelines are a mandatory requirement for annotation (Rottger et al., 2022).

In this work, we formalize and discuss the four main limitations of SAM (§2) and propose a novel annotation methodology to address them: Guideline-Centered Annotation Methodology (GCAM) (§3). Unlike SAM, GCAM decouples the data annotation from the class set by requiring annotators to map data instances to relevant parts of guidelines rather than directly assigning class labels. Class labels are then asso-

ciated with selected guideline parts based on the given problem formulation, defined by task instructors. This two-stage process provides fine-grained guideline information for each annotation, allowing for transparency in evaluating annotation quality and flexibility in reusing annotated data across multiple tasks that share guidelines. We evaluate GCAM through a human annotation study (§5) and machine learning experiments (§6), comparing it to SAM across several dimensions. Our results showcase that GCAM allows for a transparent evaluation of annotated datasets about the successful application of the prescriptive paradigm and enhances the consistency between data annotation and model training, enabling more fine-grained model evaluation compared to SAM. We release data, code and annotations to the research community.¹

2 Background and Motivation

We formalize SAM for the prescriptive annotation paradigm on text classification to highlight its properties and limitations compared to GCAM. Table 1 introduces the notation used in our work. According to the prescriptive paradigm (Rottger et al., 2022), in SAM annotators label data instances by following a set of detailed guidelines encoding the task. Standard practice typically dictates a preliminary stage both for annotators and guidelines: annotators should undergo a test to be assessed for eligibility (Klie et al., 2024) and guidelines should be refined iteratively through one or multiple discussion phases on identified edge cases (Caselli et al., 2020; Guzmán-Monteza, 2023; Jikeli et al., 2023; Kirk et al., 2023).

In SAM, the labeling consists of mapping each data instance to one or more labels of the provided

¹<https://github.com/nlp-unibo/gcam>.

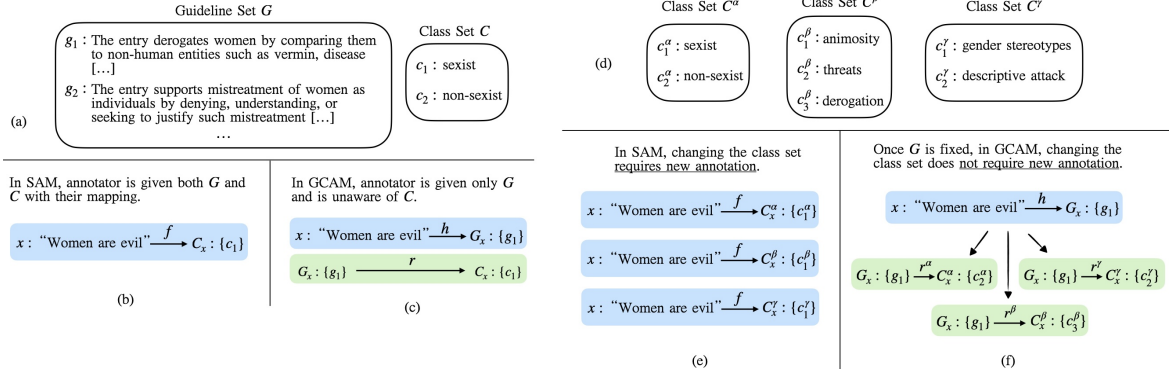


Figure 1: **(a)** Example of guideline set $\mathcal{G}$ and class set $\mathcal{C}$ for hate speech detection taken from Kirk et al. (2023). **(b)** In SAM, the annotator knows the mapping between $\mathcal{G}$ and $\mathcal{C}$ and maps the data instance x with the class subset $\mathcal{C}_x$ (blue box). **(c)** In GCAM, the annotator only knows $\mathcal{G}$ and maps x to $\mathcal{G}_x$ (blue box). Then, x is mapped to $\mathcal{C}_x$ via the class grounding function r relating $\mathcal{G}_x$ and $\mathcal{C}_x$ (green box). **(d)** Different class sets for the same $\mathcal{G}$. **(e)** In SAM, changing the class set requires new annotation; while **(f)** GCAM allows annotating with different class sets via their corresponding class grounding functions (green boxes) at the cost of a single human annotation stage (blue box).

class set $\mathcal{C}$, which is task-specific and may vary according to the problem formulation. Formally, given a dataset $\mathcal{X} = \{x_1, x_2, \dots, x_{|\mathcal{X}|}\}$ and a class set $\mathcal{C} = \{c_1, c_2, \dots, c_{|\mathcal{C}|}\}$, annotators map a data instance $x \in \mathcal{X}$ to a class subset $\mathcal{C}_x \subseteq \mathcal{C}$ by relying on the guideline set $\mathcal{G} = \{g_1, g_2, \dots, g_{|\mathcal{G}|}\}$. In a multi-class classification setting, the instance-specific class subset contains only one class (i.e., $\mathcal{C}_x = \{c_x\}$). The annotators are provided with the mapping between $\mathcal{G}$ and $\mathcal{C}$ to ground their annotations. The annotation can be seen as a function that maps a data instance x to its class subset $\mathcal{C}_x$:

$$f : \mathcal{X} \rightarrow \mathcal{P}(\mathcal{C}), \quad (1)$$

where $\mathcal{P}(\mathcal{C})$ is the power set of the class set (i.e. any combination of classes, such as $\{c_1\}$, $\{c_1, c_4\}$, etc.). An example of SAM is depicted in Figure 1(a,b), where an annotator implicitly uses the mapping between $\mathcal{G}$ and $\mathcal{C}$ to label x as *sexist*.

Assumptions. We cover text classification tasks for which annotation paradigms have been originally proposed and widely explored (Rottger et al., 2022). For both SAM and GCAM we assume that data annotation guidelines can be framed as a set of criteria (i.e. $\mathcal{G}$). We assume these criteria to be already evaluated and fixed throughout data annotation according to the prescriptive paradigm (Rottger et al., 2022). Criteria are used to reduce annotators’ effort and provide systematic instructions that are easy to assimilate and follow (Sabou et al., 2014; Ruggeri et al., 2023).

2.1 Limitations of SAM

In SAM, annotators do not report the guideline subset $\mathcal{G}_x$ they rely upon for annotating each data instance x . This information loss is the cause of diverse issues that undermine annotation quality.

Lack of Transparency. Evaluating annotation quality is crucial in ensuring the reliability of annotated data (Klie et al., 2024). The annotation quality can be influenced by three main factors: (i) intrinsic complexity of the task, (ii) guidelines clarity and coverage of the task, and (iii) annotators’ understanding of the guidelines. While task complexity is an inherent property of a problem, the other two factors can be addressed to enhance annotation quality. However, verifying annotators’ adherence to the guidelines with only class label annotations is challenging. This has implications at two distinct stages. First, during the annotator eligibility assessment, it is difficult to determine whether they have assigned their labels following the provided guidelines or using their own criteria. Second, during annotation, it is unclear whether errors stem from unclear guidelines or misinterpretation by the annotator.

Task-specific Annotations. SAM’s reliance on performing data annotation using a task-specific class set limits the reuse of annotated data across tasks that share the same guidelines. In particular, when a different task shares the same guidelines but employs a different class set, the previously

annotated data cannot be reused, turning another data annotation stage mandatory.

Model Adherence to Guidelines. ML tasks are typically structured to enable a model to learn the relation between input features (x) and target variables ($\mathcal{C}_x$). When training a model with annotated data, the underlying assumption is that the optimal mapping learned by the model from x to $\mathcal{C}_x$ is the one adherent to the guideline set $\mathcal{G}$. However, there are no explicit constraints that prevent the model from learning spurious correlations from data or from exploiting shortcuts (Geirhos et al., 2020). In principle, some models could leverage the textual knowledge encoded in the guideline subset $\mathcal{G}_x$ to guide learning. One possible way to leverage such knowledge is to integrate it into ML models as in informed machine learning (von Rue-den et al., 2023). This is typically achieved by operating at three distinct steps of the ML pipeline: by adding and/or modifying existing inputs, by introducing inductive biases, and by regularizing the learning process with ad-hoc constraints. However, to follow this direction, we would require a dataset that contains annotations linking data instances to the available knowledge to explicitly assess the knowledge integration process. When the guideline set $\mathcal{G}$ is the source of such knowledge, these annotations consist of the guideline subset $\mathcal{G}_x$. Unfortunately, these annotations cannot be created with SAM.

3 GCAM

Guideline-Centered Annotation Methodology (GCAM) is a novel methodology aimed at enriching data annotation by keeping track of the guidelines associated with data instances. Conversely to SAM, in GCAM, annotators do not have access to the class set $\mathcal{C}$, but only to the guideline set $\mathcal{G}$. This ensures the annotation to be *adherent to $\mathcal{G}$* and *independent from $\mathcal{C}$ by definition*.

3.1 Formalization

GCAM comprises two stages. First, annotators are provided with guidelines $\mathcal{G}$ and data $\mathcal{X}$, and are tasked with mapping each data instance x to its guideline subset $\mathcal{G}_x$. In the second phase, which does not involve annotators, one of the possible mappings between $\mathcal{G}$ and $\mathcal{C}$ is applied to associate instances with class labels. GCAM can be seen as

the composition of two functions:

$$h : \mathcal{X} \rightarrow \mathcal{P}(\mathcal{G}), \quad (2)$$

$$r : \mathcal{G} \rightarrow \mathcal{C}. \quad (3)$$

Equation (2) represents the annotator’s function that maps a data instance x to a guideline subset $\mathcal{G}_x$; Equation (3) is the *class grounding function* mapping each guideline g to the corresponding class c . Function h is a black box that describes each annotator, similar to f in Equation (1), while function r is known by design. Figure 1(c) illustrates GCAM. In the first stage, the human annotator maps $x = \text{“Women are evil”}$ to $\mathcal{G}_x = \{g_1\}$. In the second stage, $\mathcal{G}_x$ is mapped to $\mathcal{C}_x = \{\text{sexist}\}$ via r . Appendix A includes more annotation examples.

An Informal Interpretation. We provide an informal interpretation of GCAM and its positioning relative to SAM to explain it in a more intuitive way. In SAM, an annotator has to read guidelines accurately and decide how to label each data instance (x) based on what the guidelines dictate. Here, we highlight an important caveat: if an annotator only reports class labels, we don’t know anymore if and how they relied on the guidelines to choose such labels. This is because we do not know how an annotator thinks and behaves (i.e. the black box), nor can we get a glimpse of that from class labels. In other terms, we do not know if and to what extent an annotator followed the prescriptive annotation paradigm. GCAM aims to make these two aspects explicit to ensure adherence to the prescriptive annotation paradigm. This is achieved by instructing an annotator to label data instances with guidelines (g) rather than class labels (c). This is because we can always map guidelines to class labels, but not vice versa. In particular, the mapping is standardized, explicit (i.e. *the class grounding function r*), and not accessible to annotators to discourage task interpretations that differ from the one conveyed by the guidelines. We can then view data annotation as a process where an annotator first labels data instances with guidelines (the first stage, Equation (2)) and where we map guideline annotations ($\mathcal{G}_x$) to class annotations ($\mathcal{C}_x$) (the second stage, Equation (3)).

3.2 Properties

We describe three major properties of GCAM that address the issues of SAM (§2.1).

Transparent Adherence to Guidelines. A key aspect of GCAM is that annotators are unaware of the class set $\mathcal{C}$ and the class grounding function r . This enforces adherence by definition to the guidelines during the annotation process, as the annotators lack access to any other information about the task. In practice, GCAM requires guidelines not to contain any reference to $\mathcal{C}$ and r . Another essential part of the annotation procedure is the discussion phase, as it aims to reach an agreement on edge cases. In GCAM, discussions revolve exclusively around the guideline subset $\mathcal{G}_x$ and do not involve any reasoning on $\mathcal{C}$, as annotators lack access to it during the annotation phase. In this setting, the convergence of annotators’ points of view to the one encoded by the guidelines $\mathcal{G}$ can be effectively evaluated by inspecting $\mathcal{G}_x$.

Different Task Annotations. GCAM enables using the annotated dataset for different task annotations that share common guidelines. These annotations may stem from a taxonomy (Kirk et al., 2023) or account for discrepancies in how a task should be addressed, as it is often the case in subjective tasks (Antici et al., 2024). Consider an example with three different class sets $\mathcal{C}^\alpha, \mathcal{C}^\beta, \mathcal{C}^\gamma$ corresponding to three possible formulations of the hate speech detection task (Figure 1(d)). In GCAM (Figure 1(f)), a human annotator annotates the data instance x with guideline subset $\mathcal{G}_x$. The resulting $\mathcal{G}_x$ is mapped to the three class subsets $\mathcal{C}_x^\alpha, \mathcal{C}_x^\beta$, and $\mathcal{C}_x^\gamma$ via their corresponding class grounding functions at the cost of a single annotation process. This is a key advantage of GCAM over SAM, in which changing the class set would require new annotation stages (Figure 1(e)).

Model Adherence to Guideline. Datasets created following GCAM explicitly report the knowledge encoded in the guidelines $\mathcal{G}$. This allows ML models to be effectively trained to leverage this information. We highlight three main advantages of training a model on GCAM datasets. First, we effectively prevent the model from learning spurious correlations between x and the class subset $\mathcal{C}_x$. To learn a spurious correlation between x and $\mathcal{C}_x$, a model needs access to $\mathcal{C}_x$. However, in GCAM a model does not have access to $\mathcal{C}_x$. Therefore, a model cannot learn any sort of correlation (spurious included) between x and $\mathcal{C}_x$. Note that nothing forbids a model from learning spurious correlations between x and $\mathcal{G}_x$ instead, since $\mathcal{G}_x$ is

available to the model. Second, the decoupling between $\mathcal{G}$ and the class set $\mathcal{C}$ allows for using the model for different task annotations, as its predictions can be mapped to different class sets relying on the corresponding class grounding functions. Third, we can evaluate the model’s performance by assessing its alignment with $\mathcal{G}$ by analyzing disparities between predicted and ground-truth $\mathcal{G}_x$, thereby fostering a deeper comprehension of the model.

4 Related Work

Annotation paradigms have been formalized for data annotation to encode one shared view, such as the prescriptive paradigm (Rottger et al., 2022), or different beliefs about a phenomenon, such as the descriptive (Rottger et al., 2022), and perspectivist paradigms (Cabitza et al., 2023; Frenda et al., 2025). The goal of annotation paradigms is to regulate data annotation for a specific intended use of the collected data. In the prescriptive paradigm, this objective is achieved using annotation guidelines (Benikova et al., 2014; Sabou et al., 2014; Zaghouani et al., 2014; Krishna et al., 2023). Notable examples address subjective tasks, such as hate speech (Salminen et al., 2019; Sap et al., 2022; Davani et al., 2023), stance detection (Luo et al., 2020; AlDayel and Magdy, 2021), sentiment analysis (Waseem and Hovy, 2016; Kenyon-Dean et al., 2018), and subjectivity detection (Antici et al., 2024). In all these approaches, annotators do not keep track of the annotation guidelines they rely upon to annotate data instances. Therefore, although of crucial importance, the mapping between guidelines and data instances gets lost.

To the best of our knowledge, the only work that designs a data annotation methodology enforcing explicit grounding to annotation guidelines is (Jikeli et al., 2023), where annotators must specify which guideline applies to each data instance regarding antisemitism detection. Our work differs in several aspects. First, the work of Jikeli et al. is limited to a specific issue within social media content analysis and does not present a general formulation of the annotation process. Second, their annotation methodology relies on reporting both $\mathcal{G}_x$ and $\mathcal{C}_x$ for each data instance x . This process suffers from an annotation shortcut where annotators may first define $\mathcal{C}_x$ and subsequently $\mathcal{G}_x$, thus reversing the intended annotation order. Furthermore, the annotation methodology is tight to

a specific class set $\mathcal{C}$ and has increased annotation workload due to reporting both $\mathcal{G}_x$ and $\mathcal{C}_x$, which is not desired.

4.1 Iterative Guideline Refinement

Most related work focuses on guideline refinement rather than on a transparent annotation process adherent to guidelines. For example, [Jikeli et al. \(2023\)](#) emphasize iterative guideline development through structured discussions with annotators, mostly to surface normative assumptions and resolve contentious edge cases in hate speech annotation. Their approach foregrounds deliberation and annotator reflexivity but treats guidelines primarily as descriptive documents rather than structured, operational artifacts. [Caselli et al. \(2020\)](#) and [Guzmán-Monteza \(2023\)](#) focus more on the resolution of subjective cases, with the former proposing a decision tree guideline structure, and the latter using a supervising figure that would help with identifying subjective cases and iteratively refining the guidelines. GCAM differs from these approaches by explicitly encoding guideline structure, decision points, and revisions within the annotation process itself, rather than treating guidelines as external or static references. This enables guideline-conditioned analysis of annotations, more fine-grained attribution of disagreement to specific guideline criteria, and clearer reporting of how interpretive choices shape annotation outcomes.

GCAM transparency to guidelines might be preferable when moving from one iteration to another to limit re-annotation costs. In particular, information about the guideline subset $\mathcal{G}_x$ helps identify samples that change labeling across iterations, possibly derived from their inherent ambiguity, as it is the case in subjectivity detection ([Antici et al., 2024](#)). In this way, researchers can track how a task evolves through guideline refinement and similarly extract insights about annotation to how data cartography provides for models ([Swayamdipta et al., 2020](#)).

5 GCAM vs SAM in Annotation

We design a human annotation study to empirically compare GCAM with SAM. In particular, we consider the challenging task of subjectivity detection ([Riloff and Wiebe, 2003](#)).

Setup. We follow the annotation methodology and guidelines of [Antici et al. \(2024\)](#) and instruct

four annotators for sentence-level subjectivity detection. Annotators label a sentence as subjective or objective ($\mathcal{C} = \{\text{SUBJ}, \text{OBJ}\}$) using a set of 12 guidelines (i.e. $|\mathcal{G}| = 12$). We split the four annotators into two pairs: p_{SAM} annotates following SAM while p_{GCAM} annotates following GCAM. We use existing guidelines for SAM², while we build a variant of these guidelines for GCAM to exclude information about $\mathcal{C}$ and r . We collect 200 training instances from NewsSD-ENG ([Antici et al., 2024](#)) and split them into two data batches, b_1 and b_2 , each with 100 instances. These instances are sentences extracted from online news articles about controversial topics. We organize the 200 instances into two batches to estimate the ongoing effort required by each annotation methodology in terms of annotator workload. We instruct annotators to label instances individually, without accessing the original document from which they were extracted. To account for this, we organize batches to minimize the number of instances that belong to the same document in a batch. Both annotator pairs (i.e. p_{SAM} and p_{GCAM}) annotate the batches b_1 and b_2 , totaling 200 instances per pair. We compare GCAM and SAM both quantitatively by computing inter-annotator agreement (IAA) and qualitatively by reporting annotator feedback and analyzing annotator discussion.

Quantitative Results. We compute the IAA over reported annotations ($\mathcal{C}_x$ for p_{SAM} and $\mathcal{G}_x$ for p_{GCAM}) measured as Krippendorff’s alpha ([Krippendorff, 2004](#)) for each annotator pair. Regarding $\mathcal{C}_x$, the IAA in b_1 is 54.00 for SAM and 48.18 for GCAM, while in b_2 it is 62.00 for SAM and 54.00 for GCAM.³ These observations show that IAA is comparable for the same couple across batches and different couples on the same batch and that annotating with GCAM is comparable to SAM.⁴ Regarding $\mathcal{G}_x$, the IAA is 29.28 in b_1 and 38.74 in b_2 , highlighting that agreeing on a broad set of guidelines is notably harder. Moreover, we estimate the annotation effort by measuring the amount of time required to annotate a batch. On average, p_{SAM} takes ~ 60 min to annotate a batch, while p_{GCAM} takes ~ 150 min. As expected, GCAM has time overhead for annotation. This is due to the requirement for annota-

² $\{g_1, \dots, g_5\}$ for SUBJ and $\{g_6, \dots, g_{12}\}$ for OBJ.

³Additional IAA measurements are in Appendix B.

⁴We refer to Krippendorff’s alpha intervals.

tors to report $\mathcal{G}_x$, which is usually far more fine-grained than $\mathcal{C}_x$ and, thus, more time-consuming to assimilate. The annotation effort in GCAM is required to enforce a transparent annotation that allows evaluating annotators’ convergence to $\mathcal{G}$, as dictated by the prescriptive paradigm. We elaborate in the following paragraph.

Qualitative Results. We collect annotators’ feedback during each batch to evaluate their adherence to the annotation task; p_{SAM} annotators report relying more on their subjective criteria and less on $\mathcal{G}$ throughout the study. Specifically, annotators consult $\mathcal{G}$ mainly for challenging data instances, while they often address easier ones without referring to $\mathcal{G}$. This does not necessarily indicate a deviation from $\mathcal{G}$, as their internalized knowledge from training may still align with $\mathcal{G}$. Nonetheless, in SAM we cannot further analyze annotators’ adherence to $\mathcal{G}$. In contrast, p_{GCAM} annotators always rely on $\mathcal{G}$ by design, allowing for a direct inspection of the selected $\mathcal{G}_x$. These observations raise a major point of discussion: **to what extent does a high IAA on $\mathcal{C}_x$ reflect a correct application of the annotation paradigm?** We argue that it is hardly ever satisfactory enough. In particular, we analyze disagreement concerning $\mathcal{G}_x$ to demonstrate that (i) the majority of cases still agree on $\mathcal{C}_x$ and (ii) there is more than one reason why annotators disagree, highlighting different types of edge cases.⁵ We observe 115 cases of disagreement in 200 annotations. 66 out of them ($\sim 57\%$) agree on $\mathcal{C}_x$ while reporting disjoint $\mathcal{G}_x$ sets (58) or partially overlapping ones (8). In the remaining 49 cases, 47 are due to disjoint $\mathcal{G}_x$ sets and 2 to partially overlapping ones. After discussion, 72 cases are resolved by selecting one annotator’s $\mathcal{G}_x$, 15 by taking the union of the sets, and 25 by agreeing on a different $\mathcal{G}_x$. These results also underscore the difficulty of achieving consensus on a shared belief, as the prescriptive paradigm dictates, even when a transparent annotation like GCAM is enforced. These considerations can only be taken when providing information about $\mathcal{G}_x$. While GCAM does provide such information, SAM does not. This is a major advantage of GCAM over SAM when analyzing data quality.

⁵We apply majority voting on $\mathcal{C}_x$ to obtain a single prediction.

6 Model Alignment with Guidelines

We assess model performance on data annotated following SAM and GCAM, and we discuss the differences in error analysis when models report $\mathcal{C}_x$ (in SAM) or $\mathcal{G}_x$ (in GCAM). Our research questions are: **(Q1)** How do models perform when trained or prompted on data annotated with GCAM? **(Q2)** How do these models compare against their respective counterparts that use data annotated with SAM? **(Q3)** Which information can we gather from GCAM annotations compared to those obtained with SAM? To answer these questions, we devise two experimental settings: supervised training of encoder-based transformers and zero- and few-shot prompting of LLMs.

6.1 Data

We consider two text classification datasets that report information about $\mathcal{G}_x$ in addition to $\mathcal{C}_x$ ⁶: the Online Terms of Service (ToS-100) (Ruggeri et al., 2022) and the Explainable Detection of Online Sexism (EDOS) (Kirk et al., 2023) datasets.

ToS-100 (Ruggeri et al., 2022). ToS-100 contains 100 documents ($\sim 20k$ sentences) about unfair clause detection, where a clause is labeled as either fair (negative class) or unfair (positive class). Documents are labeled according to five categories of unfairness. For each category, legal experts manually curate a set of textual descriptions providing reasons for unfairness. These descriptions are denoted as legal rationales. To create ToS-100, annotators label positive instances by reporting one or multiple legal rationales for each unfairness category. Following Ruggeri et al. (2022), we train a binary classifier for each of the five unfairness categories. Given an unfairness category, we use the corresponding set of legal rationales as the guidelines set $\mathcal{G}$ (see Appendix C).

EDOS (Kirk et al., 2023). EDOS contains $\sim 20k$ social media comments annotated for three tasks of incremental specification: sexism detection, sexism categorization, and fine-grained vectors of sexism. For the latter task, the vectors of sexism correspond to the annotation guidelines shared among all tasks. We define $\mathcal{G}$ as this set of guidelines ($|\mathcal{G}| = 11$). Unlike in ToS-100, in EDOS annotators must select exactly one guideline g for each positive instance by design.

⁶We did not collect any new annotations on these datasets to carry out our experiments.

		SAM			GCAM			SAM			GCAM		
		<i>Binary</i>	<i>Fine-grained</i>	<i>Text-based</i> ^α				<i>Mistral</i>	<i>Llama</i>	<i>Phi</i>	<i>Mistral</i>	<i>Llama</i>	<i>Phi</i>
ToS _A	$\mathcal{C}_x$	71.61±6.76	65.31±4.15	80.14 ^{*,**} _{±5.29}				57.07	62.72	64.10	0.59	36.04	37.80
	$\mathcal{G}_x$	-	18.33±5.22	37.49 ^{*,**} _{±5.28}				-	-	-	0.11	1.44	3.89
ToS _{CH}	$\mathcal{C}_x$	87.96±3.19	83.61±2.90	90.23 ^{ns,**} _{±1.84}				62.43	60.80	62.09	1.70	21.31	8.35
	$\mathcal{G}_x$	-	34.58±4.61	44.92 ^{*,**} _{±9.93}				-	-	-	0.08	3.06	2.16
ToS _{CR}	$\mathcal{C}_x$	79.02 ±2.77	67.85±5.33	78.14 ^{ns,**} _{±3.76}				55.76	56.64	59.96	2.46	27.74	25.23
	$\mathcal{G}_x$	-	17.37±3.59	27.21 ^{*,**} _{±5.36}				-	-	-	0.46	1.18	1.17
ToS _{LTD}	$\mathcal{C}_x$	75.87±2.36	73.70±1.79	80.39 ^{*,**} _{±1.40}				60.71	55.79	58.00	8.74	21.16	7.87
	$\mathcal{G}_x$	-	19.87±2.70	21.49 ^{ns} _{±1.41}				-	-	-	0.57	5.03	4.32
ToS _{TER}	$\mathcal{C}_x$	86.86 ±1.54	78.80±5.56	84.57 ^{ns,*} _{±4.94}				62.70	54.86	57.81	2.33	18.21	2.63
	$\mathcal{G}_x$	-	10.16±2.91	20.04 ^{*,**} _{±3.08}				-	-	-	0.53	0.99	1.19
EDOS	$\mathcal{C}_x$	81.74 ±0.59	71.67±2.78	79.87 ^{*,**} _{±2.24}				56.47	67.93	63.47	25.85	38.80	28.29
	$\mathcal{G}_x$	-	27.09 ±2.69	22.97 ^{*,**} _{±3.14}				-	-	-	9.97	11.27	5.02

Table 2: Model performance on ToS and EDOS datasets for encoder-based models (left) and LLMs (right). Best results are in **bold**.

^αWe report comma-separated Wilcoxon statistical significance comparing *Text-based* to *Binary* and *Fine-grained*, respectively. (-) stands for not applicable, (ns) for no significance, (*) ≤ 0.05 , (**) ≤ 0.01 .

6.2 Encoder-based Models

We compare state-of-the-art models trained on data annotated with SAM or GCAM. For ToS-100, we consider LegalBERT (Chalkidis et al., 2020), as a top-performing model on legal benchmarks (Chalkidis et al., 2022). Likewise, we consider RoBERTaHate (Barbieri et al., 2020) for EDOS.

Problem Formulation. In SAM, given an input data instance x , the model is trained to predict $\mathcal{C}_x$. The model lacks access to $\mathcal{G}$ and only receives feedback based on $\mathcal{C}_x$ supervision. In our experiment, we consider a binary classification task. In ToS, it translates to classifying a clause as unfair; in EDOS, the task is to classify a text as sexist. We denote the SAM setting as *Binary*. Regarding GCAM, we consider two classification settings: *Fine-grained* and *Text-based*. In the *Fine-grained* setting, a model considers each guideline g as a distinct class, and no textual content of g is provided. In the *Text-based* setting, we follow the general architecture of Pappas and Henderson (2019) to train a model on guideline subsets $\mathcal{G}_x$ and, therefore, evaluate the impact of guideline textual content. More precisely, given an input pair (x, g) , the model is trained to classify if x relates to g . Thus, the model is trained to perform a textual entailment task for all $g \in \mathcal{G}$ given a data instance x . In the *Text-based* setting, we tokenize and encode x and g together and feed the resulting

embedding to the classification head.⁷

Evaluation. We perform a repeated train and test evaluation setting with 10 seed runs. We split ToS-100 at the document level in training (80), validation (10), and test (10) partitions, while we use the official data partitions for EDOS. We fine-tune models using their default hyper-parameters and apply early stopping regularization on validation loss with patience equal to 5 epochs. We report macro F_1 score on both $\mathcal{C}_x$ and $\mathcal{G}_x$ to assess model performance. For GCAM, we aggregate $\mathcal{G}_x$ predictions via r to obtain the corresponding $\mathcal{C}_x$ for binary classification. See Appendix D for hardware and implementation details.

Quantitative Results. Table 2 (left) reports classification results on both datasets. Regarding the binary classification task ($\mathcal{C}_x$), *Text-based* outperforms *Binary* in two settings (ToS_A and ToS_{LTD}), while reports comparable performance in the remaining ones. In contrast, *Text-based* significantly outperforms *Fine-grained* in all datasets. Regarding $\mathcal{G}_x$, *Text-based* outperforms *Fine-grained* in all ToS settings by a notable margin while *Fine-grained* achieves better performance in EDOS. These results highlight the benefit of integrating guideline textual content during learning (Q1). Overall, empirical observations suggest that training a model on a dataset annotated with GCAM is as effective as with SAM from a performance viewpoint (Q2).

⁷We consider other variants in Appendix E.

T / P	Neg.	Pos.	T / P	Neg.	Pos.
Neg.	2710	320	Neg.	2673	357
Pos.	199	771	Pos.	182	788

(a) $\mathcal{C}_x$ for SAM. (b) $\mathcal{C}_x$ for GCAM.

T / P	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}
g_1	2680	41	159	4	5	40	98	0	0	10	9
g_2	20	22	2	3	1	24	0	0	0	0	1
g_3	34	0	119	0	1	23	24	0	0	3	1
g_4	22	1	22	7	0	140	0	0	0	0	0
g_5	19	2	18	0	2	14	2	0	0	0	0
g_6	7	2	3	4	0	166	0	0	0	0	0
g_7	30	0	49	0	0	0	40	0	0	0	0
g_8	12	0	5	0	0	1	0	0	0	0	0
g_9	4	0	8	0	0	0	2	0	0	0	0
g_{10}	12	0	2	0	0	3	0	0	0	3	1
g_{11}	16	1	37	0	0	4	6	0	0	3	6

(c) $\mathcal{G}_x$ for GCAM.

$\mathcal{C}_x / \mathcal{G}_x$	✗	✓
✗	539	/
✓	423	3038

(d) Grounding Error Types.

Figure 2: **(a, b)** Confusion matrices of RoBERTa-Hate on $\mathcal{C}_x$ for SAM (*Binary*) and GCAM (*Text-based*). *Neg.* and *Pos.* indicate negative and positive class. **(c)** Confusion matrix on $\mathcal{G}_x$ for GCAM. **(d)** Grounding error types matrix for GCAM. ✗ and ✓ stand for wrong and correct prediction. Color coding refers to *Edge*, *Ideal*, and *Confounder* cases.

Qualitative Results. We inspect *Binary* for SAM and *Text-based* for GCAM as the top-performing models to assess how the two annotation methodologies differ when analyzing model predictions.⁸ We focus on the EDOS dataset for simplicity of evaluation since task formulation for GCAM is cast to a multi-class classification problem. Regarding SAM, the confusion matrix of *Binary* for $\mathcal{C}_x$ provides details about classification errors (Figure 2a) on the class set $\mathcal{C}$. Most errors correspond to false positives (320; compared to 199 false negatives). Compared to SAM, GCAM offers a more fine-grained assessment of model errors by providing three distinct evaluation tools. First, we compute the confusion ma-

trix on $\mathcal{C}_x$ as done in SAM (Figure 2b). We observe a similar trend to that of the SAM *Binary* model, where most errors are false positives. Second, we compute the confusion matrix on $\mathcal{G}_x$ to assess classification errors between guidelines g (Figure 2c). For instance, we can observe that g_6 is a strong confounder for g_4 instances (140), as g_3 is for g_1 (159). Third, we analyze how guidelines g are mapped to classes c to categorize prediction errors in a confusion matrix (Figure 2d). We denote this matrix as the *grounding error types matrix*. In particular, we identify three cases: $\mathcal{C}_x$ and $\mathcal{G}_x$ are incorrect (*Edge*); $\mathcal{C}_x$ and $\mathcal{G}_x$ are correct (*Ideal*); $\mathcal{C}_x$ is correct but $\mathcal{G}_x$ is incorrect (*Confounder*).⁹ While SAM aggregates all correct predictions on $\mathcal{C}_x$, GCAM distinguishes between *Ideal* and *Confounder*. This is an important characteristic of GCAM, which allows for considerations on which model predictions should be labeled as correct (Q3). In particular, we argue that *Confounder* predictions should be labeled as errors as they represent a clear violation of the prescriptive paradigm since these cases denote a disagreement on guidelines. This observation implies GCAM’s practical advantages over SAM that are not limited to a more fine-grained model evaluation: we can elaborate on which guidelines $g \in \mathcal{G}$ lead to disagreement (either on human or machine-produced annotations), fostering a targeted refinement of guidelines. We elaborate more on this aspect by performing an error analysis.

Error Analysis. Two NLP researchers with expertise in hate speech detection evaluate the *Confounder* cases regarding the *Text-based* model on the EDOS dataset. We consider a sample of 100 instances from the EDOS test set. Each researcher evaluates 50 instances. Most of the errors are associated with g_6 (casual use of gendered slurs, profanities, and insults). However, in the instances where g_6 is predicted, the loaded language (e.g., slurs, profanities, and insults) that occurs in these instances is not used in a casual way. This suggests that the model might be erroneously labeling these instances as g_6 due to a misinterpretation of the context in which such loaded language occurs. For example, the instance

“Globalist war mongering bitch, if she feels that strong about going to war

⁸We consider the best-performing seed of each model for a fair comparison.

⁹The case where $\mathcal{G}_x$ is correct and $\mathcal{C}_x$ is incorrect is not possible.

then, let her lead the charge”

contains the slur ‘*bitch*’. Although it can be non-offensive, it is used in an offensive way here. This highlights the difficulty models face in distinguishing between casual and targeted use of slurs, intentional or unintentional, which remains a significant challenge for accurate classification. Another frequently misclassified guideline is g_3 (descriptive attacks that portray women in a derogatory light —through negative generalizations about their abilities, appearance, behavior, or character, including comparison with men). The model often confuses g_3 for g_7 (immutable gender differences and stereotypes). Both categories commonly involve generalizations about gender roles (e.g., “real men”, “typical women”) and include criticism or mockery of feminists. This suggests that g_3 is often predicted in instances where the underlying intent aligns more closely with g_7 , namely, reinforcing systemic discrimination or ideological gender beliefs, albeit the expression takes the form of descriptive attacks typical of g_3 . For instance, the sentence

“Yup, 51% of the population (women) are allowed to hate on the other 49% (men), who are somehow defined as the majority...”

can be confused with g_3 because of the comparison with men that occurs in the definition of that guideline. The confusion between such guidelines is confirmed by the fact that when g_7 is the most predicted class, the most frequent true label is g_3 . Although preliminary, this error analysis underscores the added advantage of GCAM over SAM in assessing annotations —either provided by humans or by models— in relation to guidelines. In conclusion, clear and precise guidelines are essential to distinguish between these closely related categories (Q3).

6.3 LLM Prompting

We evaluate LLMs in the same setting of §6.2. We experiment with three open-source state-of-the-art LLMs: Mistral3 (Jiang et al., 2023), Llama3 (Grattafiori et al., 2024), and Phi3-mini (Abdin et al., 2024).

Problem Formulation. In SAM, given a data instance x , an LLM is prompted to predict C_x . We include $\mathcal{G}$ in the prompt to provide the task

description. In GCAM, given x , an LLM is prompted to predict $\mathcal{G}_x$. In this case, no information about r nor $\mathcal{C}$ is provided in the prompt.

Setup. We prompt LLMs in a zero-shot setting to evaluate the impact of the two problem formulations. We set the temperature to zero to exclude variations in the generated response. In this way, a change in the answer is attributed solely to the prompt definition. Moreover, we also perform in-context learning (Dong et al., 2024) by integrating examples in the prompt to guide the models. We select two examples for positive and negative classes, respectively.¹⁰

Quantitative Results. Table 2 (right) reports zero-shot classification performance. Regarding the binary classification task (C_x), we observe a notable performance degradation in the GCAM setting compared to the SAM one (Q2). Some possible motivations may be that these models are fine-tuned on instructions that require C_x rather than $\mathcal{G}_x$ as an output (Wang et al., 2022; Blevins et al., 2023) or that the classification task on $\mathcal{G}_x$ is relatively complex for models that are not explicitly aimed at classification. Therefore, prompt examples compliant with GCAM could fall into the out-of-domain boundary (Q1). Moreover, LLMs significantly underperform to encoder-based models on all tasks, with a gap of around ~ 15 -20 percentage F_1 points. This result is aligned with recent insights about LLMs on information extraction tasks, where fine-tuned encoder-based models are yet to be surpassed (Zhou et al., 2024). Adding demonstrations to guide LLMs yields mixed results (Table 3). Regarding SAM, we observe a notable performance drop (more than 20 F_1 score) in all tasks with respect to the zero-shot setting. In contrast, in GCAM few-shot prompting has a different effect depending on the model. For instance, Mistral reports substantial improvements across all tasks, Llama shows a consistent performance degradation, while Phi’s performance is sensitive to the chosen dataset. These observations align with recent findings about LLMs’ sensitivity to demonstration choice and ordering (Lu et al., 2022). Given this sensitivity, we believe that a comprehensive study of LLM prompting deserves a separate study. We provide additional considerations regarding the performance gap between

¹⁰Refer to our code repository for details about the prompts.

LLMs and encoder-based models in the following.

Qualitative Results. We inspect LLM-generated responses to better assess the performance gap observed in Table 2. A possible cause of performance degradation may be attributed to models not following reported instructions, leading to hallucinations or unrelated content (Heo et al., 2025). We do not observe such a behavior in either setting. LLMs follow prompt instructions correctly, except for a few cases ($< 2.9\%$) under SAM where the Phi model responds by selecting a guideline g rather than answering as instructed.¹¹ Consequently, we analyze model predictions in GCAM as in §6.2 by computing the grounding error types matrix for all models (Q3). In EDOS, 60-71% of model predictions fall in the *Edge* case, while 17-20% belongs to the *Confounder* case. In contrast, in ToS, *Edge* cases are about 79-97% of the cases, while *Confounder* cases are less than 1% in all categories. These results further suggest that LLMs struggle to identify the right $\mathcal{G}_x$, underscoring their incapability to address the task as formulated in GCAM. Further studies concerning more advanced prompting settings are required to corroborate these observations, which we leave for future work.

6.4 Takeaways

We summarize our empirical findings as follows. Different model architectures report contrasting performance dynamics when trained or tested with annotations produced via GCAM in comparison to SAM. Encoder-based models can efficiently discriminate between classes in C even when trained only with guidelines G (Q1) and, in some cases, outperform their counterparts that only rely on C (Q2). This is not the case for LLMs, where prompting them to label data with guidelines g is notably hard (Q1) and substantially less effective compared to prompting with only class labels c (Q2).

These results have no implications of any sort on the differences in annotation quality between SAM and GCAM, nor is this our intent. More precisely, we do not aim to demonstrate the usefulness of an annotation methodology like GCAM for a particular model architecture. Indeed, there is no direct relation between an annotation paradigm,

¹¹We repeat SAM experiments without guidelines to assess their impact in Appendix F.

		SAM			GCAM		
		Mistral	Llama	Phi	Mistral	Llama	Phi
ToS _A	$\mathcal{C}_x$	33.39	28.27	53.16	18.57	26.80	27.81
	$\mathcal{G}_x$	-	-	-	1.23	7.55	2.30
ToS _{CH}	$\mathcal{C}_x$	21.51	38.77	56.09	5.60	25.59	30.44
	$\mathcal{G}_x$	-	-	-	1.10	7.32	4.27
ToS _{CR}	$\mathcal{C}_x$	38.47	17.02	32.22	33.20	20.79	2.80
	$\mathcal{G}_x$	-	-	-	1.66	0.85	1.43
ToS _{LT D}	$\mathcal{C}_x$	36.93	14.89	54.23	13.30	28.00	3.28
	$\mathcal{G}_x$	-	-	-	1.99	4.66	1.92
ToS _{TER}	$\mathcal{C}_x$	49.19	11.98	54.40	22.14	25.49	3.59
	$\mathcal{G}_x$	-	-	-	1.23	4.63	0.55
EDOS	$\mathcal{C}_x$	53.34	33.53	55.85	31.75	24.03	22.06
	$\mathcal{G}_x$	-	-	-	11.35	11.11	4.47

Table 3: Few-shot LLMs performance on ToS and EDOS datasets.

aimed at collecting resources, and model evaluation, aimed at assessing model predictions. This is because GCAM and SAM, as any other annotation paradigm, are orthogonal to who—humans or LLMs—perform the annotation. Annotation paradigms only describe how data should be annotated, with a focus on its intended purpose. Our results only state that, assuming our experimental setting and prompt formulation, the considered models perform better or worse at predicting class set C and guideline set G labels than others. Moreover, independently of model architecture, GCAM annotations are by definition more fine-grained than those of SAM, thus enabling a more comprehensive analysis of model predictions (Q3). This is because the guideline set G is more fine-grained, larger, and descriptive than any class set C by definition.

7 Discussion

We elaborate more on GCAM about its applicability as a method, relevant side effects of its usage, its extension to other paradigms, and mitigating major drawbacks.

GCAM Applicability. Our contribution is methodological. Therefore, the advantages of GCAM over SAM are of theoretical nature. In particular, the first important contribution of our work is to highlight two crucial issues of the prescriptive annotation paradigm when implemented via SAM. First, annotators may not follow guidelines, deviating from what the prescriptive paradigm dictates. Thus, there is no guaranteed adherence to the guidelines. Second, evaluating the successful application of the paradigm by

measuring IAA on the class set $\mathcal{C}$ might lead to an overestimation of dataset quality. Indeed, this claim might hide a relatively low agreement on the guidelines, confuting the real objective of the prescriptive paradigm, which is agreeing on a single belief encoded by the guidelines. Intuitively, a high IAA on $\mathcal{C}$ does not imply a high IAA on $\mathcal{G}$, while the other way around is always true. Our annotation study suggests that this situation might be the predominant one (§5). These two issues denote an **incorrect application** of the prescriptive annotation paradigm. With GCAM, our second core contribution, we address these two issues that occur in **every annotation task** that implements the prescriptive annotation paradigm through SAM for classification tasks. We do so by trading off the cost of a more time-consuming annotation process with a **correct and transparent** application of the prescriptive annotation paradigm. If defining a correct application of the prescriptive annotation paradigm is not a strong enough motivation to apply GCAM in place of SAM, we believe our work still represents a valuable methodological contribution to raise awareness of the risks concerning applications of the prescriptive annotation paradigm and, consequently, the creation of novel datasets. Our annotation study (§5) provides empirical evidence of such risks.

Adapting Guidelines from SAM to GCAM. Providing an exact estimate of the required effort to modify existing or developing guidelines to avoid including information about the class set $\mathcal{C}$ and the mapping from the guideline set $\mathcal{G}$ to $\mathcal{C}$ (i.e., r) is hard since guidelines are task-specific by definition. However, based on our case studies, the effort is relatively marginal on average. In the case of subjectivity detection, the transformation process is straightforward since it only requires removing class label information from text. In particular, the guidelines used in (Antici et al., 2024) follow already a schematic format where each guideline g is clearly distinguishable from others.¹² This is also the case for ToS and EDOS annotation guidelines. A more complex example is the guidelines used in (Haddadan et al., 2019) concerning Argument Mining. In this example, no explicit reference to a class set $\mathcal{C}$ is reported in the guidelines, nor are guidelines associated with spe-

cific class labels. However, the guidelines follow a descriptive format where identifying clear guidelines g may not be immediate.

Guideline Availability. A requirement of using GCAM for annotation is that guidelines are always paired with published annotated data and are no longer optional material. This has two important consequences. First, it fosters reproducibility since researchers can easily understand how annotators have been instructed for data collection, thus contributing to open research (Pineau et al., 2021). Second, it raises awareness on how and for what purpose data was collected, fostering responsible research and limiting wrong uses of the data. For instance, several definitions of hate speech exist, each reflecting specific cultures (Korre et al., 2025). Therefore, training models on several datasets that share the same $\mathcal{C}$, but have incompatible $\mathcal{G}$ must be avoided.

GCAM in other Annotation Paradigms. GCAM is compatible with the perspectivist approach (Cabitza et al., 2023), which does not foresee a convergence to one label: the label variations (Plank, 2022) among annotators can be captured and entailed in the collected dataset and for model developments (Mostafazadeh Davani et al., 2022; Uma et al., 2022). In particular, capturing different annotators’ perspectives by indicating which specific guideline was selected via GCAM fosters a fine-grained comprehension of annotators’ viewpoints and evaluation of collected annotations. The applicability of GCAM to other annotation paradigms not based on detailed guidelines, such as the descriptive one (Rottger et al., 2022), is not straightforward and deserves a separate study, which we leave as future work.

Mitigating GCAM Annotation Cost. The major drawback of GCAM is the added annotation cost (§5). We elaborate more on how to mitigate this drawback. Being an instance of the prescriptive paradigm like SAM, we can leverage the same suite of solutions commonly adopted to reduce annotation time. First, there exists a wide set of annotation tools to reduce annotators’ workload, such as Inception (Klie et al., 2018), Label Studio (Tkachenko et al., 2020), and Prodigy¹³, to name a few. The wide support for tasks and high customization flexibility of these tools allows for a direct integration of GCAM for col-

¹²We report both guidelines in our repository for inspection.

¹³<https://prodi.gy/>.

lecting resources. Second, automation of data annotation by leveraging machine learning models has been applied successfully in research, either in full replacement of human annotators (Tan et al., 2024) or in a cooperative fashion (Li et al., 2023). A notable example are LLMs due to their outstanding text generation capabilities in following instructions (Mishra et al., 2022; Wang et al., 2022). In particular, when tasked to label data with an instruction prompt, LLMs have been shown to match and sometimes outperform humans (He et al., 2024; Sainz et al., 2024). Automatic data annotation is an interesting research direction, for which we report preliminary findings in Section 6.3.

8 Conclusions

We analyzed the standard application of the prescriptive annotation paradigm on text classification tasks that annotates data with class labels (i.e. SAM), highlighting two crucial issues: (i) no guaranteed annotator adherence to guidelines and (ii) overestimation of the successful application of the prescriptive paradigm by measuring annotator agreement on class labels.

To address these issues, we presented GCAM, a novel annotation methodology that reports annotation guidelines rather than class labels for each data instance. The advantages of GCAM include annotation and model adherence to guidelines, finer-grained analysis of annotated data, and support for multiple task annotations that rely on the same guidelines. The trade-off of GCAM is an increased annotation time overhead that is compensated by a correct and transparent application of the prescriptive annotation paradigm and added insights about data quality and model evaluation.

We hope GCAM can foster researchers in developing high-quality benchmarks for discerning the decision-making process and the limitations of state-of-the-art models, an ever-recurring need in NLP (Abbasi-Yadkori et al., 2024). As future work, we aim to extend our analysis to other annotation paradigms like the descriptive one (Rottger et al., 2022), develop datasets from scratch using GCAM and, consequently, further exploring model evaluation via $\mathcal{G}_x$ supervision.

Limitations

Method Applicability. We base our work and formalization of the prescriptive annotation

paradigm on text classification tasks, where we assume a class set $\mathcal{C}$ and a guideline set $\mathcal{G}$. While Rottger et al. (2022) state that the prescriptive annotation paradigm applicability is general (i.e. to any NLP task), their work discusses text classification tasks only. We do not consider other tasks (e.g., text generation, such as summarization) because, to the best of our knowledge, we are not aware of applications of the paradigm in these settings.

Annotation Study. First, we only consider one dataset. Second, we rely on a limited group of annotators and data instances due to budget and time constraints. A broader evaluation of SAM and GCAM, including other datasets, tasks of different nature and challenge, more annotators of diverse backgrounds (e.g., cultural), and more data batches, would provide more robust estimations of GCAM added effort.

Model Experiments. First, we only consider two datasets. While ToS-100 and EDOS datasets are the only datasets we found that met our criteria (i.e. having C_x and G_x annotations and a publicly available guideline set G), we acknowledge that experimenting on other datasets would broaden our empirical evaluation. Second, we only evaluate three LLMs. Although open-source and widely adopted, the landscape of LLMs is ever-growing and characterized by rapid improvements. Moreover, due to hardware constraints, all selected LLMs in our experiments undergo a 4-bit quantization process, which inevitably accounts for some performance loss. For these reasons, additional LLMs could be employed. Additionally, prompt sensitivity also plays a crucial role in LLM evaluation, which we do not explore since we use a single and fixed prompt template. Nonetheless, we remark that these additions would only aim to gather insights about how a specific model architecture (i.e. LLMs) performs on a classification task that uses G_x annotations rather than C_x . The lack of such experiments does not limit the generality of our contribution concerning the risks of applying the prescriptive annotation paradigm via SAM.

Acknowledgments

The work of F. Ruggeri is partially supported by the project European Commission’s NextGeneration EU programme, PNRR – M4C2 – Inves-

timento 1.3, Partenariato Esteso, PE00000013 - “FAIR - Future Artificial Intelligence Research” – Spoke 8 “Pervasive AI” and by the European Union’s Justice Programme under Grant Agreement No. 101087342 for the project “Principles Of Law In National and European VAT”. The work of K. Korre has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.

References

- Yasin Abbasi-Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvári. 2024. [To believe or not to believe your LLM: iterative prompting for estimating epistemic uncertainty](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Marah I Abidin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat S. Behl, Alon Benham, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. [Phi-3 technical report: A highly capable language model locally on your phone](#). *CoRR*, abs/2404.14219.
- Abeer AlDayel and Walid Magdy. 2021. [Stance detection on social media: State of the art and trends](#). *Inf. Process. Manag.*, 58(4):102597.
- Francesco Antici, Federico Ruggeri, Andrea Galassi, Katerina Korre, Arianna Muti, Alessandra Bardi, Alice Fedotova, and Alberto Barrón-Cedeño. 2024. [A corpus for sentence-level subjectivity detection on English news articles](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 273–285, Torino, Italia. ELRA and ICCL.
- Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. [TweetEval: Unified benchmark and comparative evaluation for tweet classification](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.
- Darina Benikova, Chris Biemann, and Marc Reznicek. 2014. [NoSta-D named entity annotation for German: Guidelines and dataset](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 2524–2531, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Terra Blevins, Hila Gonen, and Luke Zettlemoyer. 2023. [Prompting language models for linguistic structure](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6649–6663, Toronto, Canada. Association for Computational Linguistics.
- Federico Cabitza, Andrea Campagner, and Valerio Basile. 2023. [Toward a perspectivist turn in ground truthing for predictive computing](#). In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on*

Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023, pages 6860–6868. AAAI Press.

- Tommaso Caselli, Valerio Basile, Jelena Mitrović, Inga Kartoziya, and Michael Granitzer. 2020. [I feel offended, don't be abusive! Implicit/explicit messages in offensive and abusive language](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6193–6202, Marseille, France. European Language Resources Association.
- Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. [LEGAL-BERT: The muppets straight out of law school](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.
- Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022. [LexGLUE: A benchmark dataset for legal language understanding in English](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330, Dublin, Ireland. Association for Computational Linguistics.
- Aida Mostafazadeh Davani, Mohammad Atari, Brendan Kennedy, and Morteza Dehghani. 2023. [Hate speech classifiers learn normative social stereotypes](#). *Transactions of the Association for Computational Linguistics*, 11:300–319.
- Mark Díaz, Ian D. Kivlichan, Rachel Rosen, Dylan K. Baker, Razvan Amironesei, Vinodkumar Prabhakaran, and Emily Denton. 2022. [Crowdworksheets: Accounting for individual and collective identities underlying crowdsourced dataset annotation](#). In *FAccT '22: 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, June 21 - 24, 2022*, pages 2342–2351. ACM.
- Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. [A survey on in-context learning](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA. Association for Computational Linguistics.
- Simona Frenda, Gavin Abercrombie, Valerio Basile, Alessandro Pedrani, Raffaella Panizzon, Alessandra Teresa Cignarella, Cristina Marco, and Davide Bernardi. 2025. [Perspectivist approaches to natural language processing: a survey](#). *Language Resources and Evaluation*, 59(2):1719–1746.
- Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna M. Wallach, Hal Daumé III, and Kate Crawford. 2021. [Datasheets for datasets](#). *Commun. ACM*, 64(12):86–92.
- Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. [Shortcut learning in deep neural networks](#). *Nat. Mach. Intell.*, 2(11):665–673.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The Llama 3 herd of models. In *Neural Information Processing Systems*. Curran Associates.
- Yudi Guzmán-Monteza. 2023. [Assessment of an annotation method for the detection of spanish argumentative, non-argumentative, and their components](#). *Telematics and Informatics Reports*, 11:100068.
- Shohreh Haddadan, Elena Cabrio, and Serena Villata. 2019. [Yes, we can! mining arguments in 50 years of US presidential campaign debates](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4684–4690, Florence, Italy. Association for Computational Linguistics.
- Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2024. [AnnoLLM: Making large language models to be better crowdsourced annotators](#). *Association for Computational Linguistics*, pages 165–190.
- Juyeon Heo, Christina Heinze-Deml, Oussama Elachqar, Kwan Ho Ryan Chan, Shirley You

- Ren, Andrew C. Miller, Udhyakumar Nallasamy, and Jaya Narain. 2025. [Do LLMs "know" internally when they follow instructions?](#) In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *CoRR*, abs/2310.06825.
- Gunther Jikeli, Sameer Karali, Daniel Miehl, and Katharina Soemer. 2023. [Antisemitic messages? A guide to high-quality annotation and a labeled dataset of tweets](#). *CoRR*, abs/2304.14599.
- Kian Kenyon-Dean, Eisha Ahmed, Scott Fujimoto, Jeremy Georges-Filteau, Christopher Glasz, Barleen Kaur, Auguste Lalande, Shruti Bhandari, Robert Belfer, Nirmal Kanagasabai, Roman Sarrazingendron, Rohit Verma, and Derek Ruths. 2018. [Sentiment analysis: It's complicated!](#) In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1886–1895, New Orleans, Louisiana. Association for Computational Linguistics.
- Hannah Kirk, Wenjie Yin, Bertie Vidgen, and Paul R  ttger. 2023. [SemEval-2023 Task 10: Explainable detection of online sexism](#). In *Proceedings of the The 17th International Workshop on Semantic Evaluation, SemEval@ACL 2023, Toronto, Canada, 13-14 July 2023*, pages 2193–2210. Association for Computational Linguistics.
- Jan-Christoph Klie, Michael Bugert, Beto Boudlosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. [The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation](#). In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico. Association for Computational Linguistics.
- Jan-Christoph Klie, Richard Eckart de Castilho, and Iryna Gurevych. 2024. [Analyzing Dataset Annotation Quality Management in the Wild](#). *Computational Linguistics*, 50(3):817–866.
- Katerina Korre, Arianna Muti, Federico Ruggeri, and Alberto Barr  n-Cede  o. 2025. [Untangling hate speech definitions: A semantic componential analysis across cultures and domains](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3184–3198, Albuquerque, New Mexico. Association for Computational Linguistics.
- Klaus Krippendorff. 2004. Measuring the reliability of qualitative text analysis data. *Quality and quantity*, 38:787–800.
- Kalpesh Krishna, Erin Bransom, Bailey Kuehl, Mohit Iyyer, Pradeep Dasigi, Arman Cohan, and Kyle Lo. 2023. [LongEval: Guidelines for human evaluation of faithfulness in long-form summarization](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1650–1669, Dubrovnik, Croatia. Association for Computational Linguistics.
- Minzhi Li, Taiwei Shi, Caleb Ziems, Min-Yen Kan, Nancy Chen, Zhengyuan Liu, and Diyi Yang. 2023. [CoAnnotating: Uncertainty-guided work allocation between human and large language models for data annotation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1487–1505, Singapore. Association for Computational Linguistics.
- Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. [Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.
- Yiwei Luo, Dallas Card, and Dan Jurafsky. 2020. [Detecting stance in media on global warming](#).

- In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3296–3315, Online. Association for Computational Linguistics.
- Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. [Cross-task generalization via natural language crowdsourcing instructions](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 3470–3487. Association for Computational Linguistics.
- Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. [Dealing with disagreements: Looking beyond the majority vote in subjective annotations](#). *Transactions of the Association for Computational Linguistics*, 10:92–110.
- Nikolaos Pappas and James Henderson. 2019. [GILE: A generalized input-label embedding for text classification](#). *Trans. Assoc. Comput. Linguistics*, 7:139–155.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. [Pytorch: An imperative style, high-performance deep learning library](#). In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.
- Amandalynne Paullada, Inioluwa Deborah Raji, Emily M. Bender, Emily Denton, and Alex Hanna. 2021. [Data and its \(dis\)contents: A survey of dataset development and use in machine learning research](#). *Patterns*, 2(11):100336.
- Kenneth Peng, Arunesh Mathur, and Arvind Narayanan. 2021. [Mitigating dataset harms requires stewardship: Lessons from 1000 papers](#). In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.
- Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Hugo Larochelle. 2021. [Improving reproducibility in machine learning research\(a report from the neurips 2019 reproducibility program\)](#). *J. Mach. Learn. Res.*, 22:164:1–164:20.
- Barbara Plank. 2022. [The “problem” of human label variation: On ground truth in data, modeling and evaluation](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014. [Linguistically debatable or just plain wrong?](#) In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 507–511, Baltimore, Maryland. Association for Computational Linguistics.
- Mahima Pushkarna, Andrew Zaldivar, and Oddur Kjartansson. 2022. [Data cards: Purposeful and transparent dataset documentation for responsible AI](#). In *FACCT ’22: 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, June 21 - 24, 2022*, pages 1776–1826. ACM.
- Deborah Raji, Anna Lauren Hoffmann, Nyaleng Moorosi, Vinay Prabhu, Jake Metcalf, and Sherry Stanley. 2020. Panel discussion: Harms from AI research. In *neurIPS Workshop on Navigating the Broader Impacts of AI Research* <https://nbair.com>.
- Ellen Riloff and Janyce Wiebe. 2003. [Learning extraction patterns for subjective expressions](#). In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 105–112.
- Paul Rottger, Bertie Vidgen, Dirk Hovy, and Janet Pierrehumbert. 2022. [Two contrasting data annotation paradigms for subjective NLP tasks](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

- Technologies*, pages 175–190, Seattle, United States. Association for Computational Linguistics.
- Laura von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, Michal Walczak, Jochen Garcke, Christian Bauckhage, and Jannis Schuecker. 2023. [Informed machine learning – a taxonomy and survey of integrating prior knowledge into learning systems](#). *IEEE Transactions on Knowledge and Data Engineering*, 35(1):614–633.
- Federico Ruggeri, Francesco Antici, Andrea Galassi, Katerina Korre, Arianna Muti, and Alberto Barrón-Cedeño. 2023. [On the definition of prescriptive annotation guidelines for language-agnostic subjectivity detection](#). In *Proceedings of Text2Story - Sixth Workshop on Narrative Extraction From Texts held in conjunction with the 45th European Conference on Information Retrieval (ECIR 2023), Dublin, Ireland, April 2, 2023*, volume 3370 of *CEUR Workshop Proceedings*, pages 103–111. CEUR-WS.org.
- Federico Ruggeri, Francesca Lagioia, Marco Lippi, and Paolo Torroni. 2022. [Detecting and explaining unfairness in consumer contracts through memory networks](#). *Artif. Intell. Law*, 30(1):59–92.
- Marta Sabou, Kalina Bontcheva, Leon Derczynski, and Arno Scharl. 2014. [Corpus annotation through crowdsourcing: Towards best practice guidelines](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 859–866, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. [Gollie: Annotation guidelines improve zero-shot information-extraction](#). *International Conference on Learning Representations*.
- Joni Salminen, Hind Almerikhi, Ahmed Mohamed Kamel, Soon-gyo Jung, and Bernard J. Jansen. 2019. [Online hate ratings vary by extremes: A statistical analysis](#). In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval, CHIIR ’19*, page 213–217, New York, NY, USA. Association for Computing Machinery.
- Marta Sandri, Elisa Leonardelli, Sara Tonelli, and Elisabetta Jezek. 2023. [Why don’t you do it right? Analysing annotators’ disagreement in subjective tasks](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2428–2441, Dubrovnik, Croatia. Association for Computational Linguistics.
- Sebastin Santy, Jenny Liang, Ronan Le Bras, Katharina Reinecke, and Maarten Sap. 2023. [NLPositionality: Characterizing design biases of datasets and models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9080–9102, Toronto, Canada. Association for Computational Linguistics.
- Maarten Sap, Swabha Swayamdipta, Laura Vianna, Xuhui Zhou, Yejin Choi, and Noah A. Smith. 2022. [Annotators with attitudes: How annotator beliefs and identities bias toxic language detection](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5884–5906, Seattle, United States. Association for Computational Linguistics.
- Elena Savinova and Fermin Moscoso Del Prado. 2023. [Analyzing subjectivity using a transformer-based regressor trained on naïve speakers’ judgements](#). In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 305–314, Toronto, Canada. Association for Computational Linguistics.
- Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. [Dataset cartography: Mapping and diagnosing datasets with training dynamics](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9275–9293. Association for Computational Linguistics.

- Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansoor Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. [Large language models for data annotation and synthesis: A survey](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 930–957, Miami, Florida, USA. Association for Computational Linguistics.
- Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. 2020. [Label Studio: Data labeling software](#). Open source software available from <https://github.com/HumanSignal/label-studio>.
- Alexandra N. Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2022. [Learning from disagreement: A survey](#). *J. Artif. Int. Res.*, 72:1385–1470.
- Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbahn. 2024. Position: Will we run out of data? limits of LLM scaling based on human-generated data. In *Forty-first International Conference on Machine Learning*.
- Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krma Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Sidhartha Mishra, Sujana Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. [Super-naturalinstructions: Generalization via declarative instructions on 1600+ NLP tasks](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5085–5109. Association for Computational Linguistics.
- Zeera Waseem and Dirk Hovy. 2016. [Hateful symbols or hateful people? predictive features for hate speech detection on Twitter](#). In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California. Association for Computational Linguistics.
- Wajdi Zaghrouani, Behrang Mohit, Nizar Habash, Ossama Obeid, Nadi Tomeh, Alla Rozovskaya, Noura Farra, Sarah Alkuhlani, and Kemal Oflazer. 2014. [Large scale Arabic error annotation: Guidelines and framework](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. 2025. [Data-centric artificial intelligence: A survey](#). *ACM Comput. Surv.*, 57(5).
- Lining Zhang, Simon Mille, Yufang Hou, Daniel Deutsch, Elizabeth Clark, Yixin Liu, Saad Mahamood, Sebastian Gehrmann, Miruna Cliniciu, Khyathi Raghavi Chandu, and João Sedoc. 2023. [A needle in a haystack: An analysis of high-agreement workers on MTurk for summarization](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14944–14982, Toronto, Canada. Association for Computational Linguistics.
- Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2024. [Universalner: Targeted distillation from large language models for open named entity recognition](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

A Annotation Examples

Table 4 reports examples of sexist texts that an annotator might label via SAM or GCAM (complementing the example reported in Figure 1). The examples are taken from the EDOS dataset (Kirk et al., 2023).

B Inter-Annotator Agreement

We report additional Inter-Annotator Agreement (IAA) measurements concerning the annotation study on the comparison of SAM and GCAM (§5). Regarding C_x , the IAA, measured as Krippendorff’s alpha, on all 200 instances is 57.90 for

She’s not a real woman for being so reckless.
 Bet she’d be hot if she lost all the fat.
 Non virgin females reject virgin men on sight.
 Ponies have higher standards than white women.
 That was so stupid I thought you were a female.

Table 4: Examples of sexist texts taken from EDOS (Kirk et al., 2023) dataset.

SAM and 50.97 for GCAM. Regarding $\mathcal{G}_x$, the IAA is 33.93.

Clause type	No. Unfair clauses	$ \mathcal{G} $
Arbitration (A)	106 (0.50%)	8
Unilateral change (CH)	344 (1.68%)	7
Content removal (CR)	216 (1.06%)	17
Limitation of liability (LTD)	626 (3.07%)	18
Unilateral termination (TER)	420 (2.06%)	28

Table 5: ToS-100 statistics. We report the number and percentage of unfair clauses and the number of rationales for each unfairness category.

C Data Supplementary Details

Table 5 reports statistics about ToS-100 (Ruggeri et al., 2022).

D Hardware and Implementation Details

We implemented all baselines and methods in PyTorch (Paszke et al., 2019) and PyTorch Lightning. All experiments were run on-premises, using an NVIDIA 3060Ti GPU with 8 GB dedicated VRAM. We considered the following LLM model cards: *mistralai/Mistral-7B-Instruct-v0.3*, *meta-llama/Meta-Llama-3.1-8B-Instruct* and *microsoft/Phi-3-mini-4k-instruct*. LLMs undergo a 4-bit quantization process to reduce computational overhead and fit single-GPU training. We release all of the code and data to reproduce our experiments in an MIT-licensed public repository (see footnote 1).

E Additional Text-based Architectures

In the *Text-based* setting, we consider two more architecture variants to evaluate the impact of dif-

ferent embedding strategies: *Dot* and *Concat*. We denote the *Text-based* model reported in the main paper as *Entail* for distinction. In *Dot*, we compute the dot product of the embedding pair (x, g) and feed it to the classification head. This is the original architecture of Pappas and Henderson (2019). Similarly, *Concat* replaces the dot product via embedding concatenation. Results (Table 6) show that *Entail* outperforms other variants in all tasks but one.

F SAM Without Guidelines

We repeat SAM experiments by removing guidelines from the prompt to assess model sensitivity to guidelines and observe that zero-shot LLM performance degrades significantly (Table 7). This is a desired result since the involved benchmarks are domain-specific tasks.

		<i>Dot</i>	<i>Concat</i>	<i>Entail</i>
ToS _A	$\mathcal{C}_x$	67.07 $\pm$ 6.34	66.81 $\pm$ 6.78	80.14$\pm$5.29
	$\mathcal{G}_x$	18.31 $\pm$ 8.24	19.55 $\pm$ 7.44	37.49$\pm$5.28
ToS _{CH}	$\mathcal{C}_x$	80.93 $\pm$ 4.14	86.13 $\pm$ 7.70	90.23$\pm$1.84
	$\mathcal{G}_x$	27.35 $\pm$ 2.94	18.02 $\pm$ 3.65	44.92$\pm$9.93
ToS _{CR}	$\mathcal{C}_x$	67.85 $\pm$ 5.33	72.72 $\pm$ 5.51	78.14$\pm$3.76
	$\mathcal{G}_x$	18.21 $\pm$ 3.24	19.46 $\pm$ 3.72	27.21$\pm$5.36
ToS _{LTD}	$\mathcal{C}_x$	72.82 $\pm$ 3.04	75.00 $\pm$ 1.96	80.39$\pm$1.40
	$\mathcal{G}_x$	17.12 $\pm$ 2.70	20.31 $\pm$ 3.23	21.49$\pm$1.41
ToS _{TER}	$\mathcal{C}_x$	80.44 $\pm$ 4.19	77.64 $\pm$ 4.81	84.57$\pm$4.94
	$\mathcal{G}_x$	11.15 $\pm$ 1.50	10.38 $\pm$ 1.31	20.04$\pm$3.08
EDOS	$\mathcal{C}_x$	71.10 $\pm$ 2.20	74.96 $\pm$ 2.20	79.87$\pm$2.24
	$\mathcal{G}_x$	24.70$\pm$2.07	18.19 $\pm$ 1.37	22.97 $\pm$ 3.14

Table 6: Text-based model performance on ToS and EDOS datasets. Best results are in **bold**.

Dataset	Mistral	Llama	Phi
ToS _A	41.40 ($\downarrow$ 15.67)	36.52 ($\downarrow$ 26.20)	42.02 ($\downarrow$ 22.08)
ToS _{CH}	42.85 ($\downarrow$ 19.58)	38.69 ($\downarrow$ 22.11)	44.29 ($\downarrow$ 17.90)
ToS _{CR}	42.46 ($\downarrow$ 13.30)	37.90 ($\downarrow$ 18.74)	43.33 ($\downarrow$ 16.63)
ToS _{LTD}	45.74 ($\downarrow$ 14.97)	40.84 ($\downarrow$ 14.95)	44.85 ($\downarrow$ 13.15)
ToS _{TER}	44.66 ($\downarrow$ 18.40)	40.12 ($\downarrow$ 14.74)	46.36 ($\downarrow$ 11.45)
EDOS	40.90 ($\downarrow$ 15.57)	59.72 ($\downarrow$ 08.21)	60.88 ($\downarrow$ 02.59)

Table 7: LLMs performance on ToS and EDOS datasets when no guideline information is provided in the prompt. Brackets indicate performance degradation with respect to Table 2.