

Mapping the Course for Prompt-based Structured Prediction

Matt Pauk

University of Colorado Boulder
matt.pauk@colorado.edu

Maria Leonor Pacheco

University of Colorado Boulder
maria.pacheco@colorado.edu

Abstract

Large language models (LLMs) have demonstrated strong performance in a wide-range of language tasks without requiring task-specific fine-tuning. However, they remain prone to hallucinations and inconsistencies, and often struggle with complex reasoning, in part due to the limitations of autoregressive generation. We propose to address some of these issues, particularly for structured prediction, by combining LLMs with combinatorial inference to marry the predictive power of LLMs with the structural consistency provided by inference methods. We perform exhaustive experiments in an effort to understand which prompting strategies can best estimate confidence values for downstream symbolic inference, and find that, independent of prompting strategy, incorporating symbolic inference yields more consistent and accurate predictions than prompting alone. Finally, we show that calibration and fine-tuning with structured learning objectives further increases performance on challenging tasks, highlighting that structured learning remains valuable in the era of LLMs.

1 Introduction

Prompting large language models (LLMs) has been shown to be an effective methodology for a variety of natural language processing (NLP) tasks (Lou et al., 2024). Through pre-training on massive-scale text corpora, general-purpose LLMs acquire extensive world knowledge that can be applied across tasks without specialized training (Brown et al., 2020; Wei et al., 2022b). This makes it possible to generate answers to a broad set of questions or instructions by conditioning the generation on a textual input prompt. As a result, prompting provides us with a flexible and adaptable framework for addressing new problems, where good performance can often be achieved simply by designing appropriate prompts and curating a handful

of input-output demonstrations. However, regardless of the prompting strategy used, these models remain limited by their training objectives (Radford and Narasimhan, 2018; Wei et al., 2022a; Ouyang et al., 2022), which often lead to hallucinations and struggles with more complex reasoning tasks (Ji et al., 2023; McCoy et al., 2024). One such area where these limitations become apparent is that of structured prediction.

We use the term structured prediction to describe any machine learning task that consists of predicting several individual but related components as part of some structured object (BakIr et al., 2007). These tasks are widely common in NLP, where we are often interested in parsing text into complex linguistic structures (Mann and Thompson, 1988; Lascarides and Asher, 2007; Palmer et al., 2010; Kamath and Das, 2019) or aligning language with structured knowledge (Mao et al., 2019; Pacheco and Goldwasser, 2021). Most previous work applying LLMs to structured prediction tasks treats each individual component of the structure as an independent prompting task (Roy et al., 2022) or prompts the model to produce the entire structure all at once as a sequence of tokens (Ettinger et al., 2023). However, because these strategies are solely relying on auto-regressive generation for predicting structures, they do not have a way to strictly enforce that the predicted structure is a valid one.

Recently, Mehta et al. (2024) proposed a framework for explicitly modeling dependencies by combining traditional inference algorithms with prompt-based predictions. Their approach first scores local candidate substructures by prompting LLMs and then finds the best global output via constrained optimization. They tested their framework on semantic role labeling and co-reference resolution, and showed that enforcing consistency improves performance over unconstrained prediction while guaranteeing structurally valid outputs. While these results show the relevance and poten-

tial of constrained inference for prompt-based structured prediction, it is not clear *how* we should derive these scores from prompt-based inferences. In their work, Mehta et al. (2024) simply take the raw likelihoods of generating particular responses and directly plug them into the maximum a posteriori (MAP) inference process.

Traditional structured inference algorithms rely on weighting certainty scores for the different candidate substructures and finding the best global assignment that satisfies the structural constraints of the problem. Historically, trained ad hoc models generated label scores, which would correspond to the locally or globally normalized likelihood of the candidate label based on a learned conditional distribution $P(Y|X)$ (Lafferty et al., 2001; Collins, 2002; Chang et al., 2012; Pacheco and Goldwasser, 2021). Unlike trained discriminative classifiers, prompt-based approaches do not learn a conditional distribution for the specific task, but rather estimate the probability of generating a token in a vocabulary given the sequence of tokens provided in the prompt (Xie et al., 2022). This makes deriving certainty scores for prompt-based predictions very challenging. Given the wide range of approaches for certainty estimation (Kadavath et al., 2022; Xiong et al., 2024) and calibration (Jiang et al., 2021; Tian et al., 2023) in LLMs, it is hard to determine which -if any- of them would be appropriate for structured inference algorithms. This is especially relevant given that many of these techniques have struggled to align confidence scores with actual accuracy and reliably predict failure cases (Xiong et al., 2024).

In this paper, we are interested in systematically exploring and comparing different ways to score candidate substructures for prompt-based prediction, as well as exploring learning mechanisms to align LLMs with structured objectives. To this end, we present an exhaustive study of different ways to combine the predictive power of LLMs with structured inference to enforce the structural consistency of LLM predictions. Specifically, our goal is to answer the following research questions.

1. What is the most effective way to estimate confidence scores for individual output components from LLMs for use in combinatorial inference?
2. What is the best way to fine-tune LLMs for structured prediction tasks?

We evaluated all explored methods on two challenging discourse-level structured prediction tasks: morality framing and coreference resolution, and show that inference helps performance, regardless of the strategy used to extract confidence scores. Of the confidence estimation strategies we explore, formulating the prompt as a true/false question performs the best. Additionally, we show that fine-tuning models using the global structured prediction objective leads to improved performance over fine-tuning on the individual local decisions. All the code needed to reproduce our experiments is available to the community¹.

2 Related Work

We survey related work along three directions; deep structured prediction, structured prediction with LLMs, and confidence scoring and recalibration with LLMs.

2.1 Deep structured prediction

There is a lot of previous work combining neural models with symbolic inference for structured prediction tasks. Neural networks globally normalized using structured inference have been successfully applied to sentence-level NLP tasks such as named entity recognition and dependency parsing (Chen and Manning, 2014; Weiss et al., 2015; Ma and Hovy, 2016; Lample et al., 2016; Kiperwasser and Goldberg, 2016; Malaviya et al., 2018).

When dealing with tasks that go beyond sentence-level dependencies, most prior work combines the output scores of independently trained classifiers using inference (Beltagy et al., 2014; Ning et al., 2018; Pryor et al., 2023; Leto et al., 2024) while others create ad hoc joint learning approaches for their particular tasks (Han et al., 2019; Widmoser et al., 2021).

Our work most closely resembles Pacheco and Goldwasser (2021), who propose a general framework for combining deep learning models with structured inference. However, rather than fine-tuning custom neural networks, we leverage general-purpose generative LLMs and few-shot prompting strategies.

2.2 Structured prediction with LLMs

How best to use generative LLMs for structured prediction tasks is still a largely unexplored area.

¹<https://github.com/mappauk/prompt-based-structured-inference>

Most previous work either prompts the model to make predictions on local components without any enforcement of structural constraints (Roy et al., 2022) or prompts the model to predict the entire structure at once (Ettinger et al., 2023). Along these lines, Liu et al. (2022) propose an approach that models structures as sequences of actions in an autoregressive manner, without incurring any loss.

The closest work to our proposed method is (Mehta et al., 2024), who also combine few-shot prompting with combinatorial inference. However, they do not explore different LLM confidence estimation or calibration methods. Additionally, we explore learning strategies using global structured prediction objectives for prompt-based prediction, which to the best of our knowledge has not been explored before.

2.3 Confidence/uncertainty scoring and recalibration with LLMs

There is a considerable amount of research surrounding confidence estimation in LLMs. Geng et al. (2024) provide a survey of common confidence estimation strategies, considering both white-box methods that leverage internal model layers and/or token probabilities, as well as black-box methods that rely only on generated text.

For white-box methods, Kadavath et al. (2022) show that when prompts are formulated as true or false questions or multiple choice questions, the generated token probabilities themselves can be a well-calibrated form of confidence estimation.

For black-box methods, Manakul et al. (2023) estimate confidence values by sampling a number of generations and use the consistency and variability between generations as a proxy of confidence.

Alternatively, there is a large body of work on how to prompt and/or train models to estimate their own confidence level using numerical scales (Lin et al., 2022; Xiong et al., 2024; Liu et al., 2024).

3 Prompt-based Structured Prediction

We outline a general framework for modeling any structured prediction task via prompting. To do this, we define each problem as an LLM-based factor graph Ψ with potentials $\psi_i \in \Psi$ over all possible structures Y . Each decision $\psi_i \in \Psi$ is scored using a decoder-only LLM instance ρ with parameters θ .

Let $\rho(\mathbf{x}_i, \mathbf{y}_i; \theta)$ be the score for the potential ψ_i resulting from the prompting of an LLM ρ parameterized by θ . In this way, we can find the optimal

structure $\mathbf{y} \in Y$ by performing MAP inference as:

$$\begin{aligned} \arg \max_{\mathbf{y} \in Y} \sum_{\psi_i \in \Psi} \rho(\mathbf{x}_i, \mathbf{y}_i; \theta) \\ \text{s.t. } c(\mathbf{x}_c, \mathbf{y}_c) \quad \forall c \in C \end{aligned} \quad (1)$$

where C is the set of structural constraints defined by the factor graph Ψ , and $\mathbf{x}_c, \mathbf{y}_c$ correspond to the inputs and variables relevant to the constraints.

Following a long tradition on structured prediction for NLP, we use Integer Linear Programming (ILP) to calculate $\arg \max$. Given that every MAP inference problem with discrete variables can be represented as a linear objective, ILP provides us with the utmost flexibility to represent any structured prediction problem (Roth and Yih, 2005). In addition, all Boolean functions can be compiled into a set of linear inequalities to be used as constraints in the ILP formulation (Srikumar and Roth, 2023). Importantly, we note that this framework allows ILP to be replaced by any appropriate inference algorithm for a given task, including tractable linear programming relaxations such as AD^3 (Martins et al., 2015). Our general-purpose ILP formulation is modeled after prior work (Pacheco and Goldwasser, 2021), with specific implementation details outlined in Appendix A.

4 Strategies for Scoring Sub-structures

The scores to be used with inference are obtained by prompting an LLM. However, it is unclear how best to extract probability values for classification tasks using LLM prompts (Geng et al., 2024). Unlike a classifier explicitly trained to predict output probabilities, an LLM provides a probability distribution over all the tokens in its vocabulary. Additionally, for many proprietary models, we do not even have access to the generated token probabilities, requiring a confidence estimation strategy that works using only the plain text generations. We experiment with an exhaustive list of confidence estimation methods proposed in the literature. Following Geng et al. (2024), we divide these methods into white-box and black-box confidence estimation strategies.

4.1 White-Box Methods

We experiment with several white-box confidence estimation methods, which we describe as requiring access to the token probability values for each prompt input and generation.

4.1.1 True/False Token Prediction

Inspired by results showing that LLMs are fairly well calibrated for true/false and multiple choice questions (Kadavath et al., 2022), we format the task as a true/false question and compute confidence estimation values based on the probability that the LLM generates the true token as $p(y = True|x)$.

In the multiclass case, the confidence value is computed by normalizing the probability of y being true for a particular class c over the sum of the true token probabilities for all classes as:

$$\frac{p(y = True|x_c)}{\sum_i^{|C|} p(y = True|x_{c_i})} \quad (2)$$

4.1.2 Multiple Choice

This strategy formulates the problem as a question-answering (QA) task with the possible labels given as multiple choice options. The score for a given text and label is then the probability of generating the token of the corresponding multiple choice option o as $p(y = o|x)$.

4.1.3 Generative Classification

Instead of the standard QA format used by the True/False and Multiple Choice methods, where the model is asked to predict the label given some text to classify, we can flip the problem formulation on its head and provide the label as the input prompt and estimate the label likelihood based on the probability of the model to generate the example text. This approach has been shown to improve worst-case performance and reduce output variance (Kumar et al., 2024). Following prior work, we formulate the input prompt as "{natural language label description} {text to be classified}". We then compute the score for a particular label using the following equation:

$$\frac{1}{m} \sum_i^m \sum_j^n p(x_j|y_i x_1 \dots x_{j-1}) \quad (3)$$

Where y_i represents one of m versions of the label description and $x = x_1 \dots x_n$ represents the text to be classified. Multiple versions of the language descriptions are used to reduce variance. The score calculated from Equation 3 is then normalized over all possible labels to extract a final confidence value.

4.2 Black-box Methods

For many proprietary models, the token probabilities for the input prompt and / or the generations are not publicly available. For these models, the methods covered in Section 4.1 are not viable options. Therefore, we also explore several confidence estimation options that can be performed given only the plain-text generations of the LLM.

4.2.1 Generation Sampling

This strategy uses consistency as a proxy for confidence by computing a score based on sampling generations, relying on the inherent randomness of these models. The model is prompted n times to classify the text into one of the possible labels $L = l_1, l_2, \dots, l_m$, giving us a set of n generations $G = g_1, g_2, \dots, g_n$. The number of generations that match a particular label l_i is given by $G_{l_i} = \{g \in G | g_i = l_i\}$. The score for label l_i is then calculated as $\frac{|G_{l_i}|}{n}$.

4.2.2 Verbalized Confidence

This method prompts the model to estimate its own confidence level in the answer. We use the prompting method proposed by Xiong et al. (2024), where the model is given the text to be classified and one of the possible classes, and asked to estimate its confidence level in the answer on a scale of 0-100. We use the following prompt format:

```
Question: {q}
Possible Answer: {a}
Q: How likely is the above answer to be correct? Do not elaborate on your answer or provide any explanation, answer only with the confidence value in the following format:
Confidence: [the probability of answer label to be correct (0-100), not the one you think correct, please only include the numerical number in the range of 0-100]
```

This prompt is executed several times, and the final confidence value is given as the average confidence value elicited over all generations.

5 Learning

Sections 3 and 4 describe a process to combine few-shot prompting strategies using pre-trained language models and combinatorial inference for structured prediction tasks. However, in the process described so far, there is no learning taking place, and we are solely relying on the world knowledge

contained in the parametric space. In this section, we describe several fine-tuning strategies for the structured prediction task.

5.1 Few-Shot Score Calibration

For this strategy, we train a logistic regression layer on top of the LLM scores to better calibrate them for the structured prediction task. The parameters of the LLM remain frozen and only the weights of the logistic regression layer are tuned. A separate logistic regression model ϕ_p is used for each prompting strategy p within the structured prediction problem. We experiment with two mechanisms to train these regression models.

The first, which we refer to as **Local Calibration** involves tuning each logistic regression model separately for their respective sub-problem tasks using the cross-entropy loss function:

$$L = -\log \sum_i^C y_i \log(\hat{y}_i) \quad (4)$$

Where $\hat{y}_i = \phi_p(w)$ is the output of the logistic regression model for the strategy p , that takes as input the confidence scores w , extracted using one of the LLM prompting strategies in Section 4.

Alternatively, we jointly train all models ϕ_p using the structured hinge loss (Daumé III, 2017). To compute this loss, we perform structured inference as formulated in Section 3. However, instead of using the raw confidence scores w from the LLM, we use the corresponding output of the logistic regression layer $s_i = \phi(w)_i$. The structured hinge loss can then be formulated as follows:

$$L = \max \left\{ 0, \sum_{\hat{y}_i \in \hat{y}} s_i \hat{y}_i - \sum_{y_i \in y} s_i y_i \right\} \quad (5)$$

Where $\hat{y} \in Y$ is the current result of inference and $y \in Y$ corresponds to the gold structure. We refer to this mechanism as **Global Calibration**.

5.2 Local Fine Tuning

Supervised LLM fine-tuning has proven to be an effective method to improve LLM prompting performance for a specific task (Zhang et al., 2026). For this strategy, we fine-tune an LLM on the same prompts discussed in Section 4 using the standard loss for the next token prediction:

$$L = -\sum_{t=1}^T \log(p(y_t|x; y_1, y_2, \dots, y_{t-1})) \quad (6)$$

Fine-tuning is performed for each of the local components of the structured prediction problem. After fine-tuning the LLM, inference is performed by extracting confidence values using the same prompting strategy as used for fine-tuning.

5.3 Global Fine-tuning

For this strategy, we use the same loss formulation as in the global calibration method described above. However, rather than freezing the LLM parameters, we backpropagate the structured hinge loss (Equation 5) into the LLM itself. Before global fine-tuning, we use local fine-tuning to hot-start the model parameters.

6 Evaluation

We evaluated our framework on two complex discourse-level tasks; morality framing and coreference resolution. For coreference, we use the GENIA Coreference biomedical dataset (Su et al., 2008) and the CoNLL 2012 OntoNotes dataset (Pradhan et al., 2012). For morality framing, we use the dataset released by Roy et al. (2021). All datasets considered consist exclusively of English language text.

6.1 Morality Framing in Political Tweets

This task focuses on identifying the moral attitudes that are expressed in tweets made by members of the United States Congress (Roy et al., 2021). There are two aspects to the task; the first is identifying which of the five moral foundations (Care/Harm, Fairness/Cheating, Loyalty/Betrayal, Authority/Subversion, and Purity/Degradation) are being expressed in the tweet (Haidt and Joseph, 2004; Haidt and Graham, 2007). The other aspect of the task is to identify the moral role that the entities mentioned in the tweet are playing. As an example, consider the following tweet that expresses the moral foundation **Care/Harm**:

This common-sense bill will reduce unnecessary and duplicative burdens on health care providers and patients in need of home health services

The entity "common-sense bill" expresses the role of **entity providing care**, "health care

Method	Shots	Micro F1		Macro F1		Constraint Violations	
		MF	Role	MF	Role	C1	C2
Few-shot ICL (Roy et al., 2022)	5	0.436	n/a	n/a	n/a	n/a	n/a
True/False + constr	5	0.498 <u>0.517</u>	0.444 <u>0.452</u>	0.466 <u>0.521</u>	0.378 <u>0.408</u>	1195 0	102 0
Multiple Choice + constr	2	0.457 <u>0.530</u>	0.348 <u>0.386</u>	0.447 <u>0.512</u>	0.273 <u>0.321</u>	1133 0	234 0
Generation Sampling + constr	5	0.389 <u>0.456</u>	0.387 <u>0.388</u>	0.419 <u>0.447</u>	0.276 <u>0.317</u>	1241 0	128 0
Verbalized Confidence + constr	0	0.416 <u>0.435</u>	0.261 <u>0.322</u>	0.418 <u>0.437</u>	0.180 <u>0.274</u>	1275 0	312 0
Generative Classification + constr	0	0.410 <u>0.498</u>	0.274 <u>0.295</u>	0.426 <u>0.483</u>	0.211 <u>0.245</u>	1072 0	338 0

Table 1: Results for the Morality Frames task (Roy et al., 2021) for each of the five confidence elicitation methods (Section 4), both with and without combinatorial inference. We present results only for best number of shots (0,2,5).

Model	Micro F1		Macro F1	
	MF	Role	MF	Role
GPT-J-6B Few Shot (Roy et al., 2022)	n/a	n/a	0.436	n/a
Supervised Deep Structured Pred. (Roy et al., 2021)	n/a	n/a	0.723	0.592
Llama-8B Few Shot All In One	0.415	0.374	0.456	0.357
Llama-70B Few Shot All In One	0.519	0.478	0.506	0.401
GPT-5 Few Shot All in One	0.616	0.625	0.577	0.535
Pre-trained few-shot baseline	0.459 ± 0.01	0.345 ± 0.009	0.452 ± 0.023	0.281 ± 0.038
Pre-trained few-shot baseline (+ constr)	0.529 ± 0.007	0.388 ± 0.014	0.511 ± 0.013	0.318 ± 0.015
Pre-trained Local Calibration	0.522 ± 0.017	0.478 ± 0.006	0.450 ± 0.031	0.360 ± 0.040
Pre-trained Local Calibration (+ constr)	0.649 ± 0.006	0.562 ± 0.023	0.589 ± 0.019	0.453 ± 0.035
Pre-trained Global Calibration	0.660 ± 0.024	0.594 ± 0.031	0.638 ± 0.034	0.512 ± 0.052
Local Fine-tuned baseline	0.710 ± 0.013	0.691 ± 0.024	0.681 ± 0.018	0.595 ± 0.021
Local Fine-tuned baseline (+ constr)	0.754 ± 0.005	0.731 ± 0.013	0.720 ± 0.008	0.656 ± 0.015
Local Fine-tuned + Local Calibration	0.724 ± 0.019	0.700 ± 0.025	0.695 ± 0.032	0.610 ± 0.020
Local Fine-tuned + Local Calibration (+ constr)	0.759 ± 0.009	0.727 ± 0.013	0.722 ± 0.028	0.652 ± 0.019
Local Fine-tuned + Global Calibration	0.758 ± 0.005	0.725 ± 0.008	0.722 ± 0.020	0.658 ± 0.026
Global Fine Tuning	<u>0.764</u> ± 0.009	<u>0.732</u> ± 0.011	<u>0.731</u> ± 0.023	<u>0.662</u> ± 0.018

Table 2: Results on the Morality Frames dataset (Roy et al., 2021) after fine-tuning the LLMs. The multiple choice strategy for confidence estimation is used for all methods.

providers and patients" express the role of *target of care/harm* and "duplicative burdens" express the role of *entity causing harm*. We define prompt templates for both the moral role classification and moral foundation identification subproblems. Specific details on the prompt templates used can be found in Appendix B. After prompting, we perform inference to find the best global label assignments across both subproblems, subject to:

Constraint 1: The predicted role of an entity in a tweet must align with the moral foundation predicted for the tweet.

Constraint 2: No two entities within the same tweet can be assigned the same role.

Experimental Settings We experiment with Llama-3.1-8B-Instruct (Dubey et al., 2024) and Mistral-7B-Instruct-v0.2 (Jiang et al., 2023) as our

base models. For fine-tuning experiments, we use LoRA (Hu et al., 2022) in combination with the Llama-3.1-8B-Instruct model. Details of our hyperparameter selection and hardware used can be found in Appendices C and I, respectively. We evaluated performance by reporting macro and micro F1 scores for both moral foundation and role prediction. All results are averaged over five folds.

Results The results of the few-shot prompting methods are shown in Table 1. We experiment with 0, 2, and 5 shots for each method except for the generative classification and verbalized confidence methods, which are zero-shot methods. Table 1 only contains the best result for each method using the Llama model, as it outperformed the mistral model in all strategies. The full results for both models can be found in Appendix D. Overall, we see that inference (+ constr) improves performance compared to few-shot prompting alone, regardless

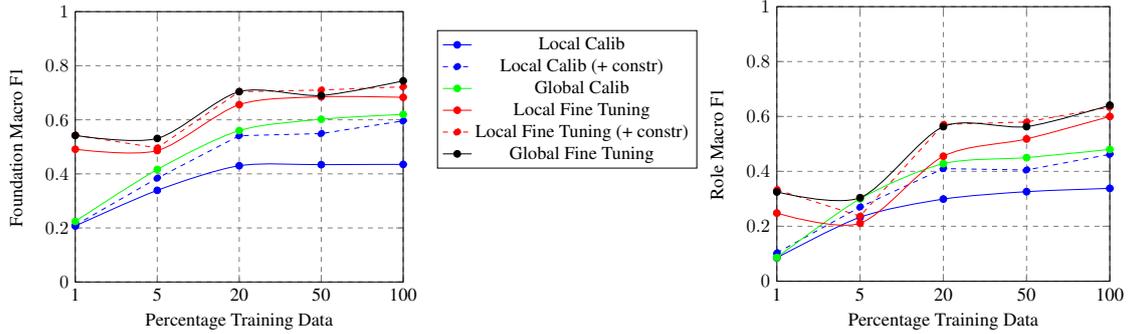


Figure 1: Morality Framing results for each training strategy over varying amounts of training data

Model	Macro F1		Violations	
	MF	Role	C1	C2
True/False - Llama-8B	0.466	0.378	1195	102
+ constr	<u>0.517</u>	<u>0.408</u>	0	0
True/False - Llama-70B	0.446	0.318	1113	50
+ constr	0.491	0.404	0	0
Multiple Choice - Llama-8B	0.447	0.273	1133	234
+ constr	<u>0.512</u>	<u>0.321</u>	0	0
Multiple Choice - Llama-70B	0.407	0.288	767	124
+ constr	0.423	0.323	0	0
Generation Sampling - Llama-8B	0.419	0.276	1241	128
+ constr	0.447	0.317	0	0
Generation Sampling - Llama-70B	0.454	0.361	1092	83
+ constr	0.494	0.399	0	0
Generation Sampling - GPT-5	0.563	0.456	735	82
+ constr	0.567	0.473	0	0

Table 3: Results on the Morality Frames dataset (Roy et al., 2021) with varying model sizes.

of the confidence estimation strategy used. The best results are achieved when using the true/false method for extracting confidence scores; this holds both before and after inference. However, the multiple choice method is comparable to true/false for the moral foundation subproblem. Furthermore, our results outperform previous prompting work on the same task (Roy et al., 2022).

Table 2 shows the results of our different calibration and fine-tuning strategies (Section 5). We use the multiple choice confidence estimation strategy for all fine-tuned models. Despite slightly worse performance than the True/False method, it requires much fewer prompts per instance, allowing for larger batch sizes during training, which are required when fine-tuning using the global structured objective (Equation 5). Similarly to the few-shot results, we find that adding inference helps performance regardless of the learning strategy. We notice that calibrating scores using logistic regression performs best when using Global Calibration. This finding holds true regardless of whether we use a pre-trained or fine-tuned LLM. However, the ad-

vantage of global calibration is much clearer in the pre-trained case. Unsurprisingly, local fine-tuning of the LLM leads to large performance gains over the few-shot version of the model.

The best performing model uses the global fine-tuning method on top of the locally fine-tuned model, outperforming the previous state-of-the-art in this task (Roy et al., 2021), which uses a classical deep structured prediction approach. We also compare our systems to different LLMs of varying size that attempt to predict the entire structure all at once with verbalized versions of our constraints in the prompt (Few Shot All In One). We find that the all-in-one prediction is an effective method that, at least in the case of Llama-70B and GPT-5, outperforms our baseline 8B models that use structured inference to enforce constraints. However, these all-in-one methods are outperformed by global calibration methods and are significantly outperformed by global fine tuning.

Table 3 displays the results of varying model sizes used to obtain priors for inference. We show results for the two best performing white-box methods (True/False and Multiple Choice) and the best performing black-box method (Generation Sampling). For each method, we show results for the Llama 8B and 70B models, and for the generation sampling method, we also experiment with the usage of GPT-5. We see that regardless of model size, the use of structured inference on top of the local predictions results in improved performance. Somewhat surprisingly, we notice that the 70B version of the Llama model actually performs slightly worse than the 8B model for the Multiple Choice and True/False methods. We suspect that the reason for this is that the additional parameters are likely encoding the ability to follow instructions for more complex tasks and, therefore, do not lead to increased performance on focused true/false or

Authority/Subversion
Rep. Greg Walden meets with C.O. veterans in Bend to discuss issues with [health care program] <i>providing_care</i> to help [rural vets] <i>being_loyal</i>
Care/Harm
Rep. Greg Walden meets with C.O. veterans in Bend to discuss issues with [health care program] <i>providing_care</i> to help [rural vets] <i>target_care</i>

Figure 2: Predictions before (top) and after (bottom) structured inference.

multiple-choice question-answering tasks.

Finally, Figure 1 shows the effects of varying amounts of training data. For the most part, the same strategies emerge as the top performers in both the low-data and high-data regimes. We notice that with very low amounts of training data (1%), all methods that fine-tune the parameter space have a larger advantage. Finally, we note that when imposing constraints (either during learning or prediction), the models converge with as little as 20% of the training data.

Error Analysis We find that some strategies are more biased than others towards certain moral foundations (see confusion matrices in Appendix E). In particular, verbalized confidence and generation sampling show a particular bias towards the authority/subversion class. In addition, we are able to find concrete evidence of structured inference correcting wrong local predictions. Consider the example shown in Figure 2, where we were able to align the predictions to be consistent with each other. That is, making sure that the moral roles assigned to entities are consistent with the overall moral foundation and that each entity plays a distinct role.

6.2 Coreference Resolution

Co-reference resolution is the task of identifying whether or not two mentions of an entity within a piece of text refer to the same entity. We break the task down into a series of subproblems, where we prompt the LLM to determine whether two entity mentions within a document are coreferent or not. Prompt template details can be found in Appendix F. We define one custom hard constraint enforcing the transitivity of coreferent entity pairs as:

Constraint 1: If entities A and B are coreferent and entities B and C are coreferent,

Method	Shots	F1	Viol.
Macaw-3B (+ constr) Mehta et al. (2024)	0	0.522	0
True/False - Mistral	5	0.815	15212
+ constr		<u>0.820</u>	0
Multiple Choice - Llama	5	0.801	24928
+ constr		<u>0.830</u>	0
Generation Sampling - Llama	5	0.834	15086
+ constr		<u>0.842</u>	0
Verbalized Confidence - Mistral	0	0.506	25872
+ constr		<u>0.512</u>	0
Generative Classification - Mistral	0	0.397	47794
+ constr		<u>0.371</u>	0

Table 4: Co-reference results for the OntoNotes dataset (Pradhan et al., 2012) across all five prompting strategies

Method	Shots	F1	Viol.
Flan-T5 (+ constr) Mehta et al. (2024)	0	0.654	0
True/False - Mistral	5	0.799	29808
+ constr		<u>0.823</u>	0
Multiple Choice - Mistral	5	0.721	72303
+ constr		<u>0.759</u>	0
Generation Sampling - Mistral	5	0.747	48570
+ constr		<u>0.781</u>	0
Verbalized Confidence - Mistral	0	0.599	123272
+ constr		<u>0.577</u>	0
Generative Classification - Llama	0	0.357	364712
+ constr		<u>0.305</u>	0

Table 5: Co-reference results for the GENIA dataset (Su et al., 2008) across all five prompting strategies.

then entities A and C must be coreferent.

Experimental Settings We experiment with Llama-3.1-8B-Instruct (Dubey et al., 2024) and Mistral-7B-Instruct-v0.2 (Jiang et al., 2023) as our base models. For fine-tuning experiments, we use LoRA (Hu et al., 2022) in combination with the Mistral-7B-Instruct-v0.2 model. Details of our hyperparameter selection and hardware use can be found in Appendices H and I, respectively. We evaluated performance by reporting macro F1 scores.

Results Few-shot results for OntoNotes and GENIA are shown in Tables 4 and 5, respectively. We experiment with both Llama and Mistral instruct models and use 0 and 5 shots for all methods except for the zero-shot methods (verbalized confidence and generative classification). We report only the best model and the best number of shots for each strategy. The full results can be found in Appendix G. We see that for all methods, except for the zero-shot ones, inference improves on the baseline results. We suspect that this is because the baseline model performance is so poor in these cases that the application of the transitivity constraint is ineffective. We find that the True/False, Multiple

Method	Macro F1
Fine-tuned Constrained Macaw-3B Mehta et al. (2024)	0.916
Pre-trained few-shot baseline	0.721
Pre-trained few-shot baseline (+ constr)	0.759
Pre-trained Local Calibration	0.771
Pre-trained Local Calibration (+ constr)	0.800
Pre-trained Global Calibration	<u>0.805</u>
Local Fine-tuned baseline	0.883
Local Fine-tuned baseline (+ constr)	0.887
Local Fine-tuned + Local Calibration	0.882
Local Fine-tuned + Local Calibration (+ constr)	0.891
Local Fine-tuned + Global Calibration	0.881
Global Fine Tuning	0.890

Table 6: Fine-tuning results on the GENIA coreference task (Su et al., 2008). The multiple choice confidence elicitation strategy is used for all models.

Choice, and Generation Sampling strategies have comparable performance for both datasets.

We test our fine-tuning strategies on the GENIA dataset and again leverage the multiple choice strategy for its balance of performance and efficiency. The results are shown in Table 6. Similarly to the morality framing task, we note that structured inference improves performance in all scenarios. However, in the fine-tuned case, it is less helpful as performance on the task is already very high.

6.3 Computational Cost Experiments

Tables 7 and 8 show the results of experiments measuring the computational efficiency of different methods when predicting a single structure for each task. We note that regardless of task, the Verbalized Confidence and Generative Classification methods are by far the most computationally expensive methods as they require multiple prompts for every candidate of a particular sub-task. This effect is even greater in the morality frames task when compared to coreference, as coreference only has two possible classes for a single subproblem, while morality frames has 16 possible moral roles for each entity and five possible moral foundations. Multiple choice is the least expensive method as it requires only a single prompt for each subproblem and is not affected by the number of classes associated with a particular subproblem. We also note that the cost of symbolic inference on top of prompting is negligible, even in the case of the most efficient prompting strategy (multiple choice).

As for the computational cost of training, the best supervised fine tuned coreference model took 13 hours and 37 minutes to train, and the best supervised fine tuned morality frames model took 5 hours and 7 minutes to train all five folds.

Method	Prompt Time (s)
True/False	12.15
Multiple Choice	1.72
Generation Sampling	12.03
Verbalized Confidence	179.42
Generative Classification	274.44

Table 7: Computational cost experiment results on the Morality Frames task (Roy et al., 2021). We measure the amount of time it takes to predict the moral frame of a tweet and the moral roles of three entities mentioned in the tweet. Regardless of strategy we find that inference adds on average 0.11 seconds.

Method	Prompt Time (s)
True/False	25.11
Multiple Choice	12.42
Generation Sampling	120.96
Verbalized Confidence	535.20
Generative Classification	321.14

Table 8: Computational cost experiment results on the GENIA coreference task (Su et al., 2008). We measure the amount of time it takes to predict coreference for all pairs of entities within a document using each strategy (154). Regardless of strategy we find that inference adds on average 3.03 seconds.

7 Conclusion

We show that structured inference is a useful tool for structured prediction tasks with LLMs, leading to consistent gains in performance over prompting alone. This holds true regardless of what strategy we use to extract confidence values from LLMs, although structuring the prompt as a true or false question works the best. Furthermore, we show that calibrating LLMs based on global structured prediction objectives can further boost performance. This finding holds true both for pre-trained models and models fine-tuned on the task. In the future, we want to explore porting the lessons learned here into multi-agent workflows, where inter-dependencies are observed, but sub-tasks are open-ended and supervision is not readily available.

Limitations

We are limited on the size of the model that we can use for our fine-tuning experimentation. We report results on 7B and 8B parameter models for Mistral and Llama, but are unable to experiment with 70B+ parameter versions of these models when

fine-tuning due to computational resource limitations. However, we experiment with larger parameter models for our pretrained, few-shot experiments (Llama-70B and GPT-5).

Lastly, we chose two representative tasks; the Morality Frames task, which provides a difficult relational reasoning task, and coreference resolution, which is a more traditional NLP task. While we believe that the evidence provided is sufficient to support our claims, future work could expand this task selection to provide greater evidence of generalization of these methods.

Ethical Considerations

To the best of our knowledge, we did not violate the ACL code of ethics during the course of our work. We use existing, public datasets for evaluation and report model hyperparameters and hardware details, as well as details of our prompting strategies to allow for the reproduction of our experiments. In addition, the code used to run our experiments has been made publicly available to the community.

Acknowledgments

This work utilized the Alpine high performance computing resource and the Blanca condo computing resource at the University of Colorado Boulder. Alpine is jointly funded by the University of Colorado Boulder, the University of Colorado Anschutz, Colorado State University, and the National Science Foundation (award 2201538). Blanca is jointly funded by computing users and the University of Colorado Boulder.

References

- Gökhan Bakır, Thomas Hofmann, Bernhard Schölkopf, Alexander J. Smola, Ben Taskar, and S.V.N. Vishwanathan. 2007. *Predicting Structured Data*. The MIT Press.
- Islam Beltagy, Katrin Erk, and Raymond Mooney. 2014. [Probabilistic soft logic for semantic textual similarity](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1210–1219, Baltimore, Maryland. Association for Computational Linguistics.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Ming-Wei Chang, Lev Ratinov, and Dan Roth. 2012. [Structured learning with constrained conditional models](#). *Mach. Learn.*, 88(3):399–431.
- Danqi Chen and Christopher Manning. 2014. [A fast and accurate dependency parser using neural networks](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar. Association for Computational Linguistics.
- Michael Collins. 2002. [Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms](#). In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 1–8. Association for Computational Linguistics.
- Hal Daumé III. 2017. *A Course in Machine Learning*. self-published.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Allyson Ettinger, Jena Hwang, Valentina Pyatkin, Chandra Bhagavatula, and Yejin Choi. 2023. [“you are an expert linguistic annotator”: Limits of LLMs as analyzers of Abstract Meaning Representation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8250–8263, Singapore. Association for Computational Linguistics.
- Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppel, Preslav Nakov, and Iryna Gurevych. 2024. [A survey of confidence estimation and calibration in large language models](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595, Mexico City, Mexico. Association for Computational Linguistics.
- Jonathan Haidt and Jesse Graham. 2007. [When morality opposes justice: Conservatives have moral intuitions that liberals may not recognize](#). *Social Justice Research*, 20(1):98–116.
- Jonathan Haidt and Craig Joseph. 2004. [Intuitive ethics: How innately prepared intuitions generate culturally variable virtues](#). *Daedalus*, 133(4):55–66.
- Rujun Han, Qiang Ning, and Nanyun Peng. 2019. [Joint event and temporal relation extraction with shared representations and structured prediction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 434–444, Hong Kong, China. Association for Computational Linguistics.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu

- Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.
- 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](#). Preprint, arXiv:2310.06825.
- Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. [How can we know when language models know? on the calibration of language models for question answering](#). *Transactions of the Association for Computational Linguistics*, 9:962–977.
- 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 Ndosue, 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](#). Preprint, arXiv:2207.05221.
- Aishwarya Kamath and Rajarshi Das. 2019. [A survey on semantic parsing](#). In *Automated Knowledge Base Construction (AKBC)*.
- Eliyahu Kiperwasser and Yoav Goldberg. 2016. [Simple and accurate dependency parsing using bidirectional LSTM feature representations](#). *Transactions of the Association for Computational Linguistics*, 4:313–327.
- Sachin Kumar, Chan Young Park, and Yulia Tsvetkov. 2024. [Gen-z: Generative zero-shot text classification with contextualized label descriptions](#). In *The Twelfth International Conference on Learning Representations*.
- John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning, ICML ’01*, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. [Neural architectures for named entity recognition](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.
- Alex Lascarides and Nicholas Asher. 2007. *Segmented Discourse Representation Theory: Dynamic Semantics With Discourse Structure*, pages 87–124. Springer Netherlands, Dordrecht.
- Alexandria Leto, Elliot Pickens, Coen Needell, David Rothschild, and Maria Leonor Pacheco. 2024. [Framing in the presence of supporting data: A case study in U.S. economic news](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 393–415, Bangkok, Thailand. Association for Computational Linguistics.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. [Teaching models to express their uncertainty in words](#). *Transactions on Machine Learning Research*.
- Shudong Liu, Zhaocong Li, Xuebo Liu, Runzhe Zhan, Derek F. Wong, Lidia S. Chao, and Min Zhang. 2024. [Can LLMs learn uncertainty on their own? expressing uncertainty effectively in a self-training manner](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21635–21645, Miami, Florida, USA. Association for Computational Linguistics.
- Tianyu Liu, Yuchen Eleanor Jiang, Nicholas Monath, Ryan Cotterell, and Mrinmaya Sachan. 2022. [Autoregressive structured prediction with language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 993–1005, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Renze Lou, Kai Zhang, and Wenpeng Yin. 2024. [Large language model instruction following: A survey of progresses and challenges](#). *Computational Linguistics*, 50(3):1053–1095.
- Xuezhe Ma and Eduard Hovy. 2016. [End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.
- Chaitanya Malaviya, Matthew R. Gormley, and Graham Neubig. 2018. [Neural factor graph models for cross-lingual morphological tagging](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2653–2663, Melbourne, Australia. Association for Computational Linguistics.
- Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. [SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages

- 9004–9017, Singapore. Association for Computational Linguistics.
- William C. Mann and Sandra A. Thompson. 1988. [Rhetorical Structure Theory: Toward a functional theory of text organization](#). *Text - Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.
- Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. 2019. [The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision](#). In *International Conference on Learning Representations*.
- A. Martins, M. A. T. Figueiredo, P. Aguiar, N. A. Smith, and E. P. Xing. 2015. [Ad3: Alternating directions dual decomposition for map inference in graphical models](#). *Journal of Machine Learning Research*, 16(Mar):495–545.
- R. Thomas McCoy, Shunyu Yao, Dan Friedman, Mathew D. Hardy, and Thomas L. Griffiths. 2024. [Embers of autoregression show how large language models are shaped by the problem they are trained to solve](#). *Proceedings of the National Academy of Sciences*, 121(41):e2322420121.
- Maitrey Mehta, Valentina Pyatkin, and Vivek Srikumar. 2024. [Promptly predicting structures: The return of inference](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 112–130, Mexico City, Mexico. Association for Computational Linguistics.
- Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018. [Joint reasoning for temporal and causal relations](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2278–2288, Melbourne, Australia. Association for Computational Linguistics.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.
- Maria Leonor Pacheco and Dan Goldwasser. 2021. [Modeling content and context with deep relational learning](#). *Transactions of the Association for Computational Linguistics*, 9:100–119.
- M.S. Palmer, D. Gildea, and N. Xue. 2010. *Semantic Role Labeling*. Synthesis lectures on human language technologies. Morgan & Claypool Publishers.
- Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. *Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes*. In *Joint conference on EMNLP and CoNLL-shared task*, pages 1–40.
- Connor Pryor, Quan Yuan, Jeremiah Liu, Mehran Kazemi, Deepak Ramachandran, Tania Bedrax-Weiss, and Lise Getoor. 2023. [Using domain knowledge to guide dialog structure induction via neural probabilistic soft logic](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7631–7652, Toronto, Canada. Association for Computational Linguistics.
- Alec Radford and Karthik Narasimhan. 2018. [Improving language understanding by generative pre-training](#).
- Dan Roth and Wen-tau Yih. 2005. [Integer linear programming inference for conditional random fields](#). In *Proceedings of the 22nd International Conference on Machine Learning, ICML ’05*, page 736–743, New York, NY, USA. Association for Computing Machinery.
- Shamik Roy, Nishanth Sridhar Nakshatri, and Dan Goldwasser. 2022. [Towards few-shot identification of morality frames using in-context learning](#). In *Proceedings of the Fifth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS)*, pages 183–196, Abu Dhabi, UAE. Association for Computational Linguistics.
- Shamik Roy, Maria Leonor Pacheco, and Dan Goldwasser. 2021. [Identifying morality frames in political tweets using relational learning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9939–9958, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Vivek Srikumar and Dan Roth. 2023. [The integer linear programming inference cookbook](#). *CoRR*, abs/2307.00171.
- Jian Su, Xiaofeng Yang, Huaqing Hong, Yuka Tateisi, and Jun’ichi Tsujii. 2008. [Coreference resolution in biomedical texts: a machine learning approach](#). *Ontologies and Text Mining for Life Sciences*, 8.
- Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. [Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.
- Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. [Finetuned language models are zero-shot learners](#). In *International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. 2015. **Structured training for neural network transition-based parsing**. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 323–333, Beijing, China. Association for Computational Linguistics.

Manuel Widmoser, Maria Leonor Pacheco, Jean Honorio, and Dan Goldwasser. 2021. **Randomized deep structured prediction for discourse-level processing**. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1174–1184, Online. Association for Computational Linguistics.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. **An explanation of in-context learning as implicit bayesian inference**. In *International Conference on Learning Representations*.

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*.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Guoyin Wang, and Fei Wu. 2026. **Instruction tuning for large language models: A survey**. *ACM Comput. Surv.*, 58(7).

Xiao Zhang, Maria Leonor Pacheco, Chang Li, and Dan Goldwasser. 2016. **Introducing DRAIL – a step towards declarative deep relational learning**. In *Proceedings of the Workshop on Structured Prediction for NLP*, pages 54–62, Austin, TX. Association for Computational Linguistics.

A ILP Formulation

We formulate the ILP objective as follows:

$$\arg \max_{\forall p \in P} \sum_j \sum_k w_{jk} \cdot p_{jk} \quad (7)$$

Here, each $p \in P$ corresponds to a particular prompting strategy, and each p_{jk} is a binary variable that represents whether a particular answer k to an instance j is true. Each w_{jk} is the probability that the corresponding p_{jk} is true. The weight itself is obtained by prompting an LLM using the prompting strategy p and extracting a confidence value, using one of the strategies discussed in Section 4.

To support the use of different prompting strategies to score the same decisions, we introduce variables d_{jk} . These variables are not used in the objective function directly, but are used in defining constraints. Similarly to previous work (Zhang et al., 2016; Pacheco and Goldwasser, 2021), we define a set of standard constraints that can be used in different structured prediction tasks.

Multi-class constraints For multiclass problems, only one variable can be activated among all possible variables for a multi-class decision on a particular instance decision.

$$\sum_k d_{jk} = 1 \quad (8)$$

Decision constraints It could be that, for a particular decision, we prompt an LLM in multiple different ways and/or with different contextual information. In order to obtain a single answer given multiple estimations, we need to constrain the assignment of the variables associated with that decision. We define a constraint to ensure that if a decision variable d_{jk} is activated, at least one of its outcome variables must be activated.

$$d_{jk} \leq \sum_p^P p_{jk} \quad (9)$$

Where P is the set of all prompting strategies. Conversely, the activation of any of the outcome variables associated with a given decision variable ensures the activation of that decision variable.

$$d_{jk} \geq \prod_p^P p_{jk} \quad (10)$$

Hard constraints We can define hard constraints to infuse domain knowledge. These constraints can be modeled in the form of “if-then” style rules. For example, let $d_{12,\text{coref}}$ be a variable that represents whether entities e_1 and e_2 are co-referent. We can ensure a transitivity constraint of the form *if $d_{12,\text{coref}}$ and $d_{23,\text{coref}}$ then $d_{13,\text{coref}}$* as:

$$d_{12,\text{coref}} + d_{23,\text{coref}} - 1 \leq d_{13,\text{coref}} \quad (11)$$

Note that this generalizes for any horn clause $d_1 \wedge d_2 \wedge \dots \wedge d_n \Rightarrow d_h$ as:

$$\sum_{i=1}^n d_i - (n - 1) \leq d_h \quad (12)$$

B Morality Frames Prompt Details

We split the Morality Frames structured prediction task into two subproblems. The prompt details for each of our prompting strategies when applied to the moral foundation classification subproblem can be found in Tables 9 and 10. The prompt details for the moral role classification subproblem can be found in Tables 11 and 12. For each subproblem we actually prompt in two different ways. In the first template we provide just the tweet itself as the [Tweet Context]. In the second template we provide the tweet, the political ideology of the tweet author, and the topic of the tweet as the [Tweet Context]. Additionally, for all strategies we provide the definitions of the associated moral foundations and roles.

C Morality Frames Hyperparameter Details

For all strategies in the few shot case we use a topk of 5 and a temperature of 0.5. For the generation sampling and verbalized confidence strategies, we sample 10 generations per each instance of a subproblem. For the generative classification method we use 10 different variations of a generation description as found in Tables 13 and 14.

For all fine-tuning strategies, we tune our hyperparameters based on the average performance on the dev split over five fold cross validation. For the logistic regression models using the pre-trained LLM model scores as input, we use a learning rate of 0.01 and batch size of 32 for both the locally and globally calibrated models. For the fine-tuned LLM model scores, our regression models use a learning rate of 0.001 and batch size of 64 when locally calibrated and a learning rate of 0.01 and batch size of 16 when globally calibrated. For the supervised fine tuning of the LLM itself, we use LoRA (Hu et al., 2022) with a rank and alpha value of 512, a batch size of 2, gradient accumulation steps of 16 and a learning rate of 2E-05. For the structured calibration we continue tuning the same LoRA weights with a batch size of 4, gradient accumulation steps of 16, and a learning rate of 2E-06.

D Extended Morality Frames Results

Tables 15 and 16 show the full results for the llama and mistral models across all five prompting strate-

gies (Section 4) on the Morality Frames structured prediction task (Roy et al., 2021).

E Error Analysis

Confusion matrices for the moral framing task can be seen in Figure 3.

F Coref Prompt Details

Tables 17 and 18 display the prompt templates used in the coreference tasks for the white box (Section 4.1) and black box (Section 4.2) prompting strategies respectively. Each strategy relies on the sentences that contain each entity mention being compared ([sent1], [sent2]) and the entity mentions themselves ([entity1], [entity2]). Additionally, the generation description strategy uses one of the variations in Table 19 as generation instructions.

G Extended Coref Results

Tables 20 and 21 show the full results for the llama and mistral models across all five prompting strategies (Section 4) on the GENIA coreference dataset (Su et al., 2008). Tables 22 and 23 show the full results on the OntoNotes coreference dataset (Pradhan et al., 2012).

H Coreference Hyperparameter Details

For all strategies in the few shot case for both the GENIA and OntoNotes coreference datasets we use a topk of 5 and a temperature of 0.5. For the generation sampling and verbalized confidence strategies, we sample 10 generations per each instance of a subproblem. For the generative classification method, we use 10 different variations of a generation description as found in Table 19.

For all fine-tuning strategies on the GENIA dataset we use the dev set for hyperparameter selection and report results on the test set. We use the same train/test split as Mehta et al. (2024). For the logistic regression models used on the pre-trained LLM scores, we use a learning rate of 0.001 and batch size of 32 for the locally calibrated models and a learning rate of 0.01 and batching done at the document level for the globally calibrated models. For the regression models trained on the fine-tuned scores, we use a learning rate of 0.01 and document batching for both locally and globally calibrated models. As with the Morality Frames task, we use LoRA for LLM fine-tuning with a rank and alpha of 512. For the supervised fine tuned model, we

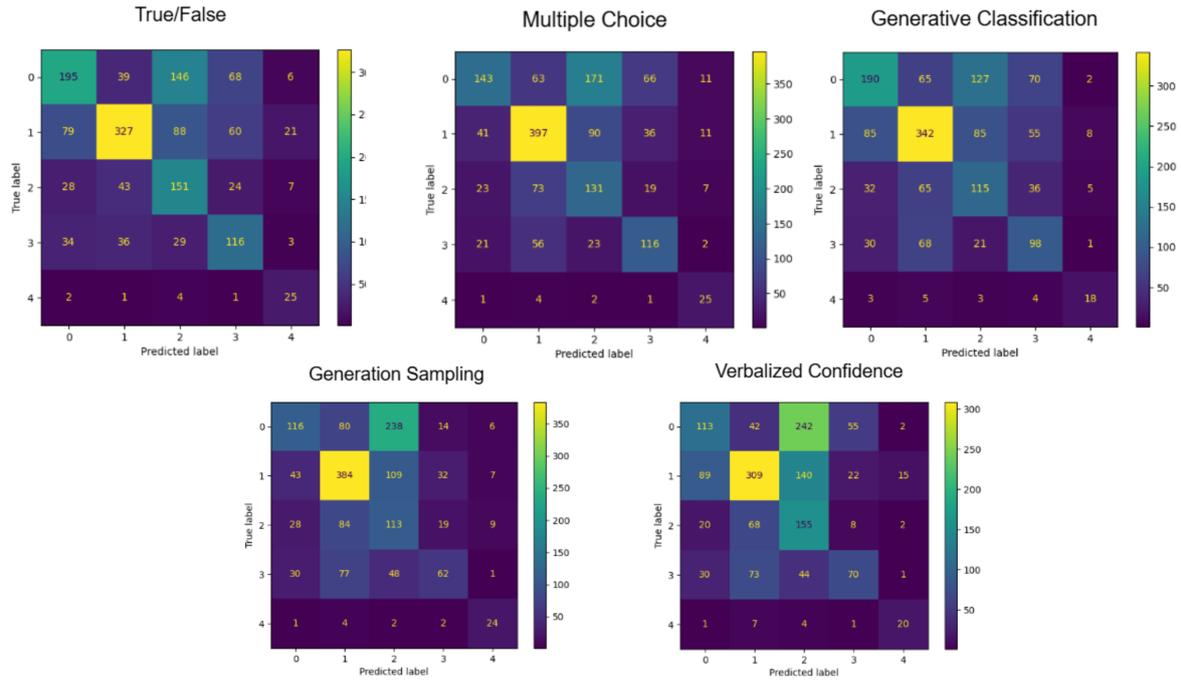


Figure 3: Confusion matrices for the morality frames task

use a batch size of 2, gradient accumulation steps of 16, and a learning rate of 2E-6. For global fine tuning, we continue tuning the same LoRA weights with a learning rate of 1E-6, a batch size of 8, and gradient accumulation steps of 32.

I Hardware Details

We use a 40GB partition of an H100 GPU for all of the 7-8B pretrained model experiments and use a full 80GB H100 for training and inference experiments with the larger models. Regardless of GPU size, all experiments utilize 8 CPUs and 32GB of CPU RAM.

Prompting Strategy	Prompt Format
True/False	<p>Consider the task of identifying the moral foundation present in a tweet from a U.S congress member. The five moral foundations and their corresponding definitions are given below:</p> <p>[Moral Frame Definitions]</p> <p>Given the moral foundations, their definitions, and the task of identifying the foundation present in a tweet, answer the following true/false question regarding whether a specific moral foundation is present in a tweet.</p> <p>[Tweet Context]</p> <p>Q. "The moral foundation expressed in the tweet is [label]." - true or false? A.</p>
Multiple Choice	<p>Consider the task of identifying the moral foundation present in a tweet from a U.S congress member. The five moral foundations and their corresponding definitions are given below:</p> <p>[Moral Frame Definitions]</p> <p>Given the moral foundations, their definitions, and the task of identifying the foundation present in a tweet, answer the following multiple choice questions regarding whether a specific moral foundation is present in a tweet. Answer only with the letter corresponding to the correct answer.</p> <p>[Tweet Context]</p> <p>Q. What moral foundation is being expressed in the given tweet?</p> <p>Choices:</p> <p>(A) CARE/HARM (B) FAIRNESS/CHEATING (C) AUTHORITY/SUBVERSION (D) PURITY/DEGRADATION (E) LOYALTY/BETRAYAL</p>
Generative Classification	<p>Consider the task of generating a tweet made by a U.S congress member given a description of the moral foundation that is being expressed in the tweet. The five moral foundations and their corresponding definitions are given below:</p> <p>[Moral Frame Definitions]</p> <p>Given the moral foundations and their definitions, generate a tweet given a description of the tweet.</p> <p>Generate a tweet based on the following description: Generation description: [Generation Description] Tweet: [Tweet]</p>

Table 9: Prompt templates for the moral foundation classification subproblem for each of the white-box strategies (Section 4.1). [Generation Description] refers to one of the generation descriptions found in Table 13.[label] is one of the five moral foundations (Care/Harm, Fairness/Cheating, Loyalty/Betrayal, Authority/Subversion, and Purity/Degradation).

Prompting Strategy	Prompt Format
Generation Sampling	<p>Consider the task of identifying the moral foundation present in a tweet from a U.S congress member. The five moral foundations and their corresponding definitions are given below:</p> <p>[Moral Frame Definitions]</p> <p>Given the moral foundations and their definitions, identify the foundation present in the given tweet. Only answer with the correct moral foundation and do not provide any justification or explanation.</p> <p>[Tweet Context]</p> <p>Q. What moral foundation is being expressed in the given tweet?</p>
Verbalized Confidence	<p>Consider the task of identifying the moral foundation present in a tweet. The five moral foundations and their corresponding definitions are given below:</p> <p>[Moral Frame Definitions]</p> <p>Given the moral foundations and their definitions and the task of identifying the foundation present in a tweet. Estimate the probability that the specified moral foundation is expressed in the tweet. Please answer with the following format: “Confidence: [the probability of answer [label] to be correct (0-100), not the one you think correct, please only include the numerical number in the range of 0-100]”</p> <p>Question: What is the moral foundation present in the following tweet: [Tweet Context]?</p> <p>Possible Answer: [label]</p> <p>Q: How likely is the above answer to be correct? Do not elaborate on your answer or provide any explanation, answer only with the confidence value in the following format:</p> <p>“Confidence: [the probability of answer [label] to be correct (0-100), not the one you think correct, please only include the numerical number in the range of 0-100]”</p>

Table 10: Prompt templates for the moral foundation classification subproblem for each of the black-box strategies (Section 4.2). [label] is one of the five moral foundations (Care/Harm, Fairness/Cheating, Loyalty/Betrayal, Authority/Subversion, and Purity/Degradation).

Prompting Strategy	Prompt Format
True/False	<p>Consider the task of identifying the moral role of an entity present in a tweet from a U.S congress member. Definitions for the five moral foundations and their associated roles are given below:</p> <p>[Moral Frame/Role Definitions]</p> <p>Given the possible moral roles, the definitions of their associated moral foundations, and the task of identifying the moral role of an entity in a tweet, answer the following true/false question regarding whether an entity is expressing a particular moral role in a tweet.</p> <p>[Tweet Context]</p> <p>Q. "The moral role of "[Entity]" expressed in the tweet is [label]." - true or false? A.</p>
Multiple Choice	<p>Consider the task of identifying the moral role of an entity present in a tweet from a U.S congress member. Definitions for the five moral foundations and their associated roles are given below:</p> <p>[Moral Frame/Role Definitions]</p> <p>Given the possible moral roles, the definitions of their associated moral foundations, and the task of identifying the moral role of an entity in a tweet, answer the following multiple choice question regarding whether an entity is expressing a particular moral role in a tweet. Answer only with the letter corresponding to the correct answer.</p> <p>[Tweet Context]</p> <p>Q. What is the moral role of "[Entity]" expressed in the given tweet?</p> <p>(A) Target of care/harm (B) Entity causing harm (C) Entity providing care (D) Target of fairness/cheating (E) Entity ensuring fairness ...</p>
Generative Classification	<p>Consider the task of generating a tweet made by a U.S congress member given a description of the moral role being expressed by an entity in the tweet. Definitions for the five moral foundations and their associated moral roles are given below:</p> <p>[Moral Frame/Role Definitions]</p> <p>Given the possible moral roles and the definitions of their associated moral foundations, generate a tweet about a given entity expressing a particular moral role.</p> <p>Generate a tweet based on the following description: Generation description: [Generation Description] Tweet: [Tweet]</p>

Table 11: Prompt templates for the moral role identification subproblem for each of the white-box strategies (Section 4.1). [Generation Description] refers to one of the generation descriptions found in Table 14. [label] is one of the sixteen possible moral roles (Target of care/harm, Entity causing harm, Entity providing care, Target of fairness/cheating, Entity ensuring fairness, Entity doing cheating, Target of loyalty/betrayal, Entity being loyal, Entity doing betrayal, Justified authority, Justified authority over, Failing authority, Failing authority over, Target of purity/degradation, Entity preserving purity, Entity causing degradation).

Prompting Strategy	Prompt Format
Generation Sampling	<p>Consider the task of identifying the moral role of an entity present in a tweet from a U.S congress member. Definitions for the five moral foundations and their associated roles are given below:</p> <p>[Moral Frame/Role Definitions]</p> <p>Given the possible moral roles and the definitions of their associated moral foundations, identify the moral role of an entity in a tweet. Only answer with the correct moral role for the entity and do not provide any justification or explanation.</p> <p>[Tweet Context]</p> <p>Q. What is the moral role of "[Entity]" expressed in the given tweet?</p>
Verbalized Confidence	<p>Consider the task of identifying the moral role of an entity present in a tweet. Definitions for the five moral foundations and their associated roles are given below:</p> <p>[Moral Frame/Role Definitions]</p> <p>Given the possible moral roles, the definitions of their associated moral foundations, and the task of identifying the moral role of an entity in a tweet. Please answer with the following format:</p> <p>“Confidence: [the probability of answer [label] to be correct (0-100), not the one you think correct, please only include the numerical number in the range of 0-100]”</p> <p>Question: What is the moral role of the entity "[Entity]" expressed in the following tweet: [Tweet Context]?</p> <p>Possible Answer: [label]</p> <p>Q: How likely is the above answer to be correct? Do not elaborate on your answer or provide any explanation, answer only with the confidence value in the following format:</p> <p>“Confidence: [the probability of answer [label] to be correct (0-100), not the one you think correct, please only include the numerical number in the range of 0-100]”</p>

Table 12: Prompt templates for the moral role identification subproblem for each of the back-box strategies (Section 4.2). [label] is one of the sixteen possible moral roles (Target of care/harm, Entity causing harm, Entity providing care, Target of fairness/cheating, Entity ensuring fairness, Entity doing cheating, Target of loyalty/betrayal, Entity being loyal, Entity doing betrayal, Justified authority, Justified authority over, Failing authority, Failing authority over, Target of purity/degradation, Entity preserving purity, Entity causing degradation).

Generation Description
"This tweet expresses the moral foundation [Moral Foundation] which is defined as: [Moral Foundation Definition]"
"This tweet reflects the moral foundation [Moral Foundation], which is defined as: [Moral Foundation Definition]"
"The tweet showcases the moral foundation [Moral Foundation], described as: [Moral Foundation Definition]"
"In this tweet, the moral foundation [Moral Foundation] is expressed, defined as: [Moral Foundation Definition]"
"This tweet highlights the moral foundation [Moral Foundation], which means: [Moral Foundation Definition]"
"The moral foundation [Moral Foundation] is conveyed in this tweet, defined as: [Moral Foundation Definition]"
"This tweet demonstrates the moral foundation [Moral Foundation], described as: [Moral Foundation Definition]"
"In this tweet, the author expresses the moral foundation [Moral Foundation], which is defined as: [Moral Foundation Definition]"
"This tweet communicates the moral foundation [Moral Foundation], described as: [Moral Foundation Definition]"
"This tweet conveys the moral foundation [Moral Foundation], defined as: [Moral Foundation Definition]"

Table 13: Generation descriptions used for the generative classification strategy on the morality foundation classification subproblem. As with [Kumar et al. \(2024\)](#), we generated these variations using ChatGPT with the following prompt: Write 10 paraphrases of this sentence as a Python list. "This tweet expresses the moral foundation [Moral Foundation] which is defined as [Moral Foundation Definition]."

Generation Description
"In this tweet, the entity [Entity] displays the moral role [Moral Role], defined as: [Moral Role Definition]"
"This tweet shows the entity [Entity] exhibiting the moral role [Moral Role], which is defined as: [Moral Role Definition]"
"The entity [Entity] in this tweet demonstrates the moral role [Moral Role], described as: [Moral Role Definition]"
"In this tweet, [Entity] reflects the moral role [Moral Role], which is defined as: [Moral Role Definition]"
"[Entity] in this tweet exemplifies the moral role [Moral Role], defined as: [Moral Role Definition]"
"This tweet portrays the entity [Entity] as embodying the moral role [Moral Role], described as: [Moral Role Definition]"
"The entity [Entity] in this tweet illustrates the moral role [Moral Role], which is defined as: [Moral Role Definition]"
"[Entity] shows the moral role [Moral Role] in this tweet, defined as: [Moral Role Definition]"
"In this tweet, [Entity] reveals the moral role [Moral Role], defined as: [Moral Role Definition]"
"This tweet features [Entity] expressing the moral role [Moral Role], which is defined as: [Moral Role Definition]"

Table 14: Generation descriptions used for the generative classification strategy on the morality role identification subproblem. As with [Kumar et al. \(2024\)](#), we generated these variations using ChatGPT with the following prompt: Write 10 paraphrases of this sentence as a Python list. "This entity [Entity] in this tweet exhibits the moral role [Moral Role] defined as [Moral Role Definition]."

Method	Shots	Micro F1		Macro F1		Constraint Violations	
		MF	Role	MF	Role	C1	C2
True/False	0	0.369	0.347	0.401	0.247	1079	330
+ constr		<u>0.429</u>	0.362	<u>0.437</u>	0.293	0	0
Multiple Choice	0	0.403	0.291	0.374	0.184	830	501
+ constr		<u>0.445</u>	<u>0.332</u>	<u>0.412</u>	<u>0.231</u>	0	0
Generation Sampling	0	0.326	<u>0.363</u>	0.317	0.239	981	172
+ constr		<u>0.371</u>	0.351	<u>0.331</u>	<u>0.263</u>	0	0
Verbalized Confidence	0	0.416	0.261	0.418	0.180	1275	312
+ constr		<u>0.435</u>	<u>0.322</u>	<u>0.437</u>	<u>0.274</u>	0	0
Generative Classification	0	0.410	0.274	0.426	0.211	1072	338
+ constr		0.498	<u>0.295</u>	0.483	<u>0.245</u>	0	0
True/False	2	0.448	0.386	0.422	0.313	1053	215
+ constr		<u>0.486</u>	0.391	<u>0.467</u>	0.346	0	0
Multiple Choice	2	0.457	0.348	0.447	0.273	1133	234
+ constr		0.530	<u>0.386</u>	0.512	<u>0.321</u>	0	0
Generation Sampling	2	<u>0.461</u>	0.354	<u>0.492</u>	0.271	1113	155
+ constr		0.451	<u>0.357</u>	0.475	<u>0.324</u>	0	0
True/False	5	0.498	0.444	0.466	0.378	1195	102
+ constr		<u>0.517</u>	0.452	0.521	0.408	0	0
Multiple Choice	5	0.451	0.319	0.454	0.219	1254	219
+ constr		0.528	<u>0.361</u>	<u>0.516</u>	<u>0.277</u>	0	0
Generation Sampling	5	0.389	0.387	0.419	0.276	1241	128
+ constr		<u>0.456</u>	<u>0.388</u>	<u>0.447</u>	<u>0.317</u>	0	0

Table 15: Full Results for the Morality Frames task (Roy et al., 2021) using Llama-3.1-8B-Instruct for each of the five confidence elicitation methods (Section 4), both with and without combinatorial inference.

Method	Shots	Micro F1		Macro F1		Constraint Violations	
		MF	Role	MF	Role	C1	C2
True/False	0	0.352	0.350	0.274	0.213	974	459
+ constr		<u>0.457</u>	<u>0.353</u>	0.352	0.206	0	0
Multiple Choice	0	0.407	0.290	0.207	<u>0.154</u>	645	400
+ constr		<u>0.415</u>	0.386	<u>0.251</u>	0.145	0	0
Generation Sampling	0	0.347	<u>0.329</u>	0.214	<u>0.195</u>	1133	306
+ constr		<u>0.382</u>	0.262	<u>0.351</u>	0.192	0	0
Verbalized Confidence	0	0.412	0.238	0.239	<u>0.136</u>	617	547
+ constr		0.430	<u>0.267</u>	<u>0.248</u>	0.132	0	0
Generative Classification	0	0.374	0.189	0.333	0.140	1307	346
+ constr		<u>0.415</u>	<u>0.217</u>	<u>0.365</u>	<u>0.183</u>	0	0
True/False	2	0.444	0.427	0.317	0.306	1332	197
+ constr		0.528	0.436	0.495	0.358	0	0
Multiple Choice	2	0.453	0.317	0.433	0.201	1159	213
+ constr		<u>0.491</u>	<u>0.367</u>	<u>0.452</u>	<u>0.252</u>	0	0
Generation Sampling	2	0.401	<u>0.354</u>	0.388	0.233	1348	158
+ constr		<u>0.440</u>	0.333	<u>0.417</u>	<u>0.252</u>	0	0
True/False	5	0.500	0.464	0.483	0.354	1214	151
+ constr		0.536	0.426	0.513	0.377	0	0
Multiple Choice	5	0.458	0.332	<u>0.433</u>	0.224	1109	228
+ constr		<u>0.498</u>	<u>0.384</u>	0.426	<u>0.246</u>	0	0
Generation Sampling	5	0.357	<u>0.368</u>	<u>0.386</u>	0.214	1464	180
+ constr		<u>0.412</u>	0.346	0.376	<u>0.230</u>	0	0

Table 16: Full Results for the Morality Frames task (Roy et al., 2021) using Mistral-7B-Instruct-v0.2 for each of the five confidence elicitation methods (Section 4), both with and without combinatorial inference.

Prompting Strategy	Prompt Format
True/False	<p>Consider the task of coreference resolution, where the goal is to identify whether or not two different entity mentions refer to the same underlying entity. Given two entity mentions and their representative sentences, answer the following true/false question regarding whether the two entity mentions refer to the same entity.</p> <p>Entity 1: [entity1] Sentence 1: [sent1] Entity 2: [entity2] Sentence 2: [sent2]</p> <p>Q. "The entity "[entity1]" mentioned in Sentence 1 and the entity "[entity2]" mentioned in Sentence 2 are [label] entities." - true or false? A.</p>
Multiple Choice	<p>Consider the task of coreference resolution, where the goal is to identify whether or not two different entity mentions refer to the same underlying entity. Given two entity mentions and their representative sentences, answer the following multiple choice question regarding whether or not the two entity mentions are coreferent or not. Answer only with the letter corresponding to the correct answer.</p> <p>Entity 1: [entity1] Sentence 1: [sent1] Entity 2: [entity2] Sentence 2: [sent2]</p> <p>Q. What is the relationship between the entity "[entity1]" mentioned in Sentence 1 and "[entity2]" mentioned in Sentence 2? (A) Coreferent (B) Distinct</p>
Generative Classification	<p>Consider the task of coreference resolution. Given two entities mentions that are either coreferent or distinct, generate two sentences each containing one of the entity mentions.</p> <p>Generate two sentences based on the following description:</p> <p>Generation Description: [Generation Description] Sentence 1: [sent1] Sentence 2: [sent2]</p>

Table 17: Coreference prompt templates for each of the white-box prompting strategies (Section 4.1). The [Generation Description] variable refers to one of the generation descriptions in Table 19. [label] is either ‘coreferent’ or ‘distinct’.

Prompting Strategy	Prompt Format
Generation Sampling	<p>Consider the task of coreference resolution, where the goal is to identify whether or not two different entity mentions refer to the same underlying entity. Given two entity mentions and their representative sentences, identify whether the entity mentions are coreferent or distinct. Answer only with "coreferent" or "distinct" and do not provide any justification or explanation.</p> <p>Entity 1: [entity1] Sentence 1: [sent1] Entity 2: [entity2] Sentence 2: [sent2]</p> <p>Q. What is the relationship between the entity "[entity1]" mentioned in Sentence 1 and "[entity2]" mentioned in Sentence 2? Answer only with "coreferent" or "distinct" and do not provide any justification or explanation.</p>
Verbalized Confidence	<p>Consider the task of coreference resolution, where the goal is to identify whether or not two different entity mentions refer to the same underlying entity. Given two entity mentions and their representative sentences, identify whether the entity mentions are coreferent or distinct. Please answer with the following format: "Confidence: [the probability that the two entity mentions are [label] (0-100), please only include the numerical number in the range of 0-100]"</p> <p>Entity 1: [entity1] Sentence 1: [sent1] Entity 2: [entity2] Sentence 2: [sent2]</p> <p>Q: How likely is it that the two entity mentions are [label]. Do not elaborate on your answer or provide any explanation, answer only with the confidence value in the following format:</p> <p>"Confidence: [the probability that the two entity mentions are [label] (0-100), please only include the numerical number in the range of 0-100]"</p>

Table 18: Coreference prompt templates for each of the black-box prompting strategies (Section 4.2). [label] is either 'coreferent' or 'distinct'.

Generation Description
"The entity mention '[entity1]' in the first sentence and the entity mention of '[entity2]' in the second sentence are [label]"
"The mention of '[entity1]' in sentence one and '[entity2]' in sentence two are considered [label]."
"'[entity1]' from the first sentence and '[entity2]' from the second sentence are labeled as [label]."
"In the first sentence, '[entity1]' is mentioned, and in the second sentence, '[entity2]' is mentioned — these are [label]."
"The entities '[entity1]' and '[entity2]', from the first and second sentences respectively, are [label]."
"We determine that '[entity1]' in sentence one and '[entity2]' in sentence two are [label]."
"There is a mention of '[entity1]' in the first sentence and of '[entity2]' in the second; they are [label]."
"According to the sentence context, '[entity1]' and '[entity2]' are identified as [label]."
"It is determined that the entity '[entity1]' in the first sentence and '[entity2]' in the second sentence are [label]."
"In the first sentence, the mention of '[entity1]', and in the second, the mention of '[entity2]', are considered [label]."

Table 19: Generation descriptions used for the generative classification strategy (Section 4.1.3). As with [Kumar et al. \(2024\)](#), we generated these variations using ChatGPT with the following prompt: Write 10 paraphrases of this sentence as a Python list. "The entity mention [entity1] in the first sentence and the entity mention of [entity2] in the second sentence are [label]."

Method	Shots	F1	Viol.
True/False	0	0.695	106502
+ constr		0.758	0
Multiple Choice	0	0.427	345664
+ constr		0.420	0
Generation Sampling	0	0.566	249450
+ constr		0.598	0
Verbalized Confidence	0	0.321	338178
+ constr		0.269	0
Generative Classification	0	0.357	364712
+ constr		0.305	0
True/False	5	0.731	89982
+ constr		0.797	0
Multiple Choice	5	0.665	131086
+ constr		0.719	0
Generation Sampling	5	0.750	55980
+ constr		0.779	0

Table 20: Full Co-reference results for the GENIA dataset (Su et al., 2008) using Llama-3.1-8B-Instruct across all five prompting strategies (Section 4).

Method	Shots	F1	Viol.
True/False	0	0.772	42054
+ constr		0.795	0
Multiple Choice	0	0.600	163160
+ constr		0.592	0
Generation Sampling	0	0.658	128512
+ constr		0.686	0
Verbalized Confidence	0	0.599	123272
+ constr		0.577	0
Generative Classification	0	0.219	259610
+ constr		0.150	0
True/False	5	0.799	29808
+ constr		0.823	0
Multiple Choice	5	0.721	72303
+ constr		0.759	0
Generation Sampling	5	0.747	48570
+ constr		0.781	0

Table 21: Full Co-reference results for the GENIA dataset (Su et al., 2008) using Mistral-7B-Instruct-v0.2 across all five prompting strategies (Section 4).

Method	Shots	F1	Viol.
True/False	0	0.763	21264
+ constr		0.774	0
Multiple Choice	0	0.476	53008
+ constr		0.455	0
Generation Sampling	0	0.661	51524
+ constr		0.657	0
Verbalized Confidence	0	0.423	44480
+ constr		0.291	0
Generative Classification	0	0.315	27062
+ constr		0.272	0
True/False	5	0.753	36884
+ constr		0.777	0
Multiple Choice	5	0.801	24928
+ constr		0.830	0
Generation Sampling	5	0.834	15086
+ constr		0.842	0

Table 22: Full Co-reference results for the OntoNotes dataset (Pradhan et al., 2012) using Llama-3.1-8B-Instruct across all five prompting strategies (Section 4).

Method	Shots	F1	Viol.
True/False	0	0.747	9540
+ constr		0.729	0
Multiple Choice	0	0.648	42444
+ constr		0.627	0
Generation Sampling	0	0.709	32312
+ constr		0.702	0
Verbalized Confidence	0	0.506	25872
+ constr		0.512	0
Generative Classification	0	0.397	47794
+ constr		0.371	0
True/False	5	0.815	15212
+ constr		0.820	0
Multiple Choice	5	0.741	32712
+ constr		0.748	0
Generation Sampling	5	0.792	21858
+ constr		0.799	0

Table 23: Full Co-reference results for the OntoNotes dataset (Pradhan et al., 2012) using Mistral-7B-Instruct-v0.2 across all five prompting strategies (Section 4).