Regression (and Scoring) Aware Inference with LLMs

Anonymous ACL submission

Abstract

Large language models (LLMs) have shown strong results on a range of applications, including regression and scoring tasks. Typically, one obtains outputs from an LLM via autoregressive sampling from the model's output distribution. We show that this inference strategy can be sub-optimal for common regression and scoring evaluation metrics. As a remedy, we build on prior work on Minimum Bayes Risk decoding, and propose alternate inference strategies for regression and scoring that estimate the Bayes-optimal solution for the given metric in closed-form from sampled responses. We show that our proposal yields significant improvements over baselines across datasets and models.

1 Introduction

Large language models (LLMs) are currently the most capable models across many NLP tasks (OpenAI et al., 2023; Google and et al., 2023; Touvron et al., 2023; Gemini Team and et al., 2023). Owing to their remarkable *few-* and *zero-shot* abilities (Wei et al., 2022; Kojima et al., 2023), pre-trained LLMs are often applied without *any* additional training on in-domain datasets: instead, one may query the LLM with a suitably crafted input prompt.

More recently, LLMs have been successfuly applied to regression and scoring tasks. For example, Gruver et al. (2023) explored zero-shot learning for time series prediction; Vacareanu et al. (2024) showed how LLMs are remarkably strong at in-context learning for regression tasks; Liu and Low (2023); Yang et al. (2023) considered the autoregressive finetuning over numerical targets applied to arithmetic tasks; and Qin et al. (2023) applied LLMs for listwise ranking.

The quality of an LLM is often assessed using an application-specific *evaluation metric*. One popular metric is the *exact match* (EM), which penalises *any* response not exactly equal to the one in the dataset annotation. This is an analogue of the conventional classification accuracy. While EM is an intuitive metric, there are many applications where it is not suitable. This is particularly true with tasks such relevance scoring (Cer et al., 2017) and sentiment analysis (Fathony et al., 2017), where the outputs are numerical or ordinal categories. In these cases, one instead prefers metrics such as the squared error, mean absolute error or ranking scores that take the ordinal nature of the outputs into account.

Despite the wide variety of evaluation metrics, LLM *inference* is typically performed in the same manner for *every* task: namely, one performs auto-regressive sampling from the LLM's underlying distribution (see §2). While intuitive, such inference does not explicitly consider the downstream evaluation metric of interest. This raises a natural question: *is there value in adapting the inference procedure to the evaluation metric at hand for regression and scoring tasks*?

A prominent line of work takes a decision-theoretic approach to the above problem. Dubbed as *Minimum Bayes Risk* (MBR) decoding, this approach seeks to optimize at inference time the metric of choice under the model's distribution (Bickel and Doksum, 1977; Kumar and Byrne, 2004; Eikema and Aziz, 2020; Bertsch et al., 2023). Much of the work on MBR is focused on evaluation metrics for machine translation and text generation tasks, such as the BLEU score. Of particular interest in this literature are self-consistency based decoding strategies that take a (weighted) majority vote of sampled responses (Wang et al., 2023a), which have shown to provide quality gains in arithmetic and reasoning problems.

In this paper, we build on the existing literature on MBR to design metric-aware inference strategies for *general regression and scoring* tasks. We first observe that choosing the most likely target for an input corresponds to *inherently optimizing for the EM* metric, and is consequently *not optimal* when EM is not the metric

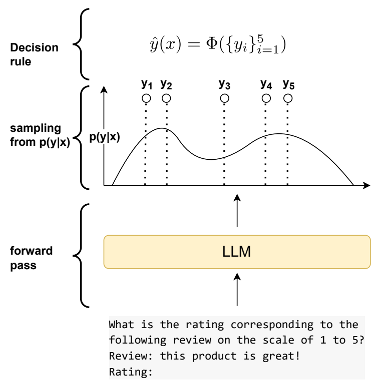


Figure 1: Illustration of the metric-aware LLM inference for regression and scoring tasks. An input x is passed to the LLM, and samples are drawn from the distribution over targets y conditioned on x. These are then used to find the target optimizing a metric m through a closed-form decision rule Φ (e.g., mean or median); Table 1 presents specific solutions across metrics.

of choice. As a remedy, we propose estimating the Bayes-optimal output for a metric under the model's distribution; we show that this admits a *closed-form* solution for common regression and ranking metrics, and only requires estimating a simple statistic from the sampled responses. In contrast, prior MBR methods for translation and summarization often require heuristically solving an intractable maximization problem (Ehling et al., 2007; Bertsch et al., 2023). We show across datasets and models how our approach yields gains over choosing the most likely target, and over self-consistency based approaches.

2 When (naïve) LLM inference fails on regression tasks

We begin with the problem setting. For a finite vocabulary V of *tokens* (e.g., words in English), let D denote a distribution over *inputs* $x \in X \subseteq V^*$ comprising of strings of tokens and *targets* $y \in Y$. Let $p(y \mid x)$ denote the conditional distribution over targets given an input. We consider a special case of this setting where $Y \subset \mathbb{R}$ corresponds to numeric targets. Here, we assume that each $y \in Y$ has a unique string representation $\mathtt{str}(y) \in V^*$; for example, the integer 1 has the string encoding "1". In a slight abuse of notation, we use $p(y \mid x) \doteq p(\mathtt{str}(y) \mid x)$ to denote the conditional probability of output y given input x.

A *language model* (LM) takes a string x as input and predicts an output $\hat{y} \in Y$. Typically, the LM first produces a distribution $\hat{p}(\cdot \mid x)$ over targets, from which a prediction is derived via a suitable *inference* (or *decoding*) procedure. Perhaps the most common inference strategy is to choose the mode of $\hat{p}(y \mid x)$:

$$\hat{y}(x) := \operatorname*{argmax}_{y \in Y} \hat{p}(y \mid x). \tag{1}$$

In practice, one may approximate the mode by employing greedy decoding or beam search, or sampling multiple candidates and picking the among them the one with the highest likelihood score (Naseh et al., 2023).

The quality of an LM's prediction is measured by some *evaluation metric* $m(y, \hat{y})$, where we assume that *higher* values are *better*. While the *exact match* (EM), given by $m(y, \hat{y}) = \mathbb{1}(y = \hat{y})$, is a commonly used evaluation metric, there are a range of other metrics popularly used to evaluate LMs. These include the (negative) squared error $m(y, \hat{y}) = -(y - \hat{y})^2$ or absolute error $m(y, \hat{y}) = -|y - \hat{y}|$ for regression tasks. A natural goal is to then choose the inference strategy $\hat{y}(x)$ to maximize the metric m of interest, i.e., to maximize the expected utility:

$$\mathbb{E}_{(x,y)\sim D}\left[m(y, \hat{y}(x))\right]. \tag{2}$$

For many choices of metric $m(y, \hat{y}(x))$, picking the mode of the predicted distribution (1) can be sub-optimal for (2).

As an example, consider the task of predicting the star rating (on the scale 1–5) associated with a review text. Suppose $m(y, \hat{y})$ is the negative absolute error between the true and predicted ratings. Given the review text "`This keybord is suitable for fast typers`", suppose the responses and the associated probabilities from an LM are {"1": 0.3, "2": 0.0, "3": 0.3, "4": 0.0, "5": 0.4}. The mode of the predicted probabilities is "5". In contrast, the maximizer of (2) is the median rating "3". We provide examples for Amazon reviews with the learned probability distributions in Figure 2 (Appendix).

3 Metric-aware LLM inference

3.1 Minimum Bayes risk decoding

We seek to design decoding strategies that maximize the expected utility in (2). Ideally, if we had access to the true conditional probabilities $p(\cdot \mid x)$, the maximizer of (2) is given by:

$$\hat{y}^*(x) \in \operatorname*{argmax}_{y' \in Y} \mathbb{E}_{y \sim p(\cdot \mid x)}\left[m(y, y')\right]. \tag{3}$$

When m is the EM metric, the optimal inference strategy is $\hat{y}^*(x) \in \operatorname{argmax}_{y \in Y} p(y \mid x)$, which is what common approaches such as greedy decoding seek to approximate.

Problem	Labels Y	Predictions	Metric	Optimal decision rule
Classification	$1,\ldots,K$	$1,\ldots,K$	$\mathbb{1}(y=\hat{y})$	$\hat{y}(x) := \text{argmax}_y\ p(y\,\|\,x)$
Regression	$\mathbb{R}$	$\mathbb{R}$	$-(y-\hat{y})^2$	$\hat{y}(x) := \mathbb{E}_{y\sim p(\cdot\,\|\,x)}[y]$
Ordinal regression	$1,\ldots,K$	$1,\ldots,K$	$-\|y-\hat{y}\|$	$\hat{y}(x) := \text{median}[p(\cdot\,\|\,x)]$
Bi-partite ranking	± 1	$\mathbb{R}$	AUC with $c_{y,y'}=1$	$\hat{y}(x) := p(y=+1\|x)$
Multi-partite ranking	$1,\ldots,K$	$\mathbb{R}$	AUC with $c_{y,y'}=\|y-y'\|$	$\hat{y}(x) := \mathbb{E}_{y\sim p(\cdot\,\|\,x)}[y]$

Table 1: Optimal decision rule for different evaluation metrics. See (6) for definition of AUC.

In general, however, the optimal decoding strategy can have a very different form, and the mode of $p(\cdot|x)$ has been shown to be suboptimal on generation tasks (Eikema and Aziz, 2020). For example, as shown in Table 1, for evaluation metrics over numerical targets such as the squared error or the absolute error, the optimal inference strategy is to simply take the mean or median of $p(\cdot|x)$ (Bishop, 2006).

3.2 Closed-form optimal solution

In practice, we mimic the Bayes-optimal solution in (3) with two approximations. First, we replace the true conditional distribution $p(\cdot\,|\,x)$ with the LM's predicted distribution $\hat{p}(\cdot\,|\,x)$. This is a reasonable approximation when the LM is pre-trained with next-token prediction objective based on the softmax cross-entropy loss; the latter is a strictly proper loss, whose minimizer under an unrestricted hypothesis class is the true conditional distribution $p(y\,|\,x)$ (Gneiting and Raftery, 2007). Second, we estimate the expectation in (3) by sampling K outputs from $\hat{p}(\cdot\,|\,x)$, and then computing:

$$\hat{y}(x) \in \underset{y' \in Y}{\text{argmax}} \sum_{i=1}^{K} m(y_i, y'). \qquad (4)$$

Even with these approximations, maximizing (4) over all outputs Y is intractable in general.

Prior literature on MBR for metrics like BLEU heuristically perform this maximization over a small set of candidates (Ehling et al., 2007; Bertsch et al., 2023). In this paper, we consider regression and scoring metrics, for which the above maximization can be computed in *closed-form*. As shown in Table 1, these solutions can be estimated by computing simple statistics from the sampled responses, such as the sample mean $\hat{y}(x) = \frac{1}{K}\sum_{i=1}^{K} y_i$ for the squared error. We refer to this approach as **R**egression (and scoring) **A**ware **I**nference with LLMs (**RAIL**).

3.3 Post-hoc temperature scaling

When sampling from $\hat{p}(\cdot\,|\,x)$, it often helps to apply a temperature scaling to the LM logits to control the diversity of the sampled outputs. This is particularly important in our procedure where we wish to approximate expectations over $\hat{p}(\cdot|x)$ using a few samples.

In practice, one may sample from $\hat{p}(\cdot\,|\,x)$ with temperature $T = 1$, and apply temperature scaling in a post-hoc manner by employing a weighted version of the objective in (4):

$$\hat{y}(x) \in \underset{y' \in Y}{\text{argmax}} \sum_{i=1}^{K} \left(\hat{p}(y_i|x)\right)^{\alpha} \cdot m(y_i, y'), \quad (5)$$

where α can be seen as the temperature scaling parameter. The above summation is a (scaled) estimate of $\mathbb{E}_{y\sim\hat{p}(\cdot\,|\,x)}\left[\hat{p}(y\,|\,x)^{\alpha} \cdot m(y,y')\right]$. For probabilities $\hat{p}(y_i\,|\,x) \propto \exp(f(x,y_i))$ defined by logits $f(x,y_i)$, this is equivalent to computing the expectation under the temperature-scaled distribution $\hat{p}_{\alpha}(y\,|\,x) \propto \exp((1+\alpha)\cdot f(x,y))$, *albeit* a normalization factor. We consider an analogous weighting scheme for the plug-in estimators of the closed-form solutions in Table 1.

3.4 Extension to multi-partite ranking

Our metric-aware decoding proposal also applies to scoring tasks, where the label space Y is discrete, e.g. $\{1,\ldots,K\}$, but we require the LLM to predict a real-valued score $\hat{y}(x) \in \mathbb{R}$ for each prompt x such that prompts with higher labels receive a higher score. One typically measures the performance of the predicted scores $\hat{y}(x)$ using a pairwise ranking metric such as AUC:

$$\text{AUC}(\hat{y}) = 1 - \mathbb{E}\left[c_{y,y'} \cdot \mathbb{1}(\hat{y}(x) < \hat{y}(x')) \,\middle|\, y > y'\right], \quad (6)$$

which penalizes the scorer $\hat{y}$ with a penalty $c_{y,y'}$ whenever it mis-ranks a pair (x,x') with $y > y'$.

Despite AUC being non-decomposable (not a summation of per-example results), Uematsu and Lee (2015) show that when the costs are the difference between the labels, i.e., $c_{y,y'} = |y-y'|$, the optimal scorer admits a closed-form solution, and is given by the expected label under distribution $p(\cdot|x)$: $\hat{y}^*(x) = \mathbb{E}_{y\sim p(\cdot|x)}[y]$. One can thus readily apply our RAIL approach to estimate this solution from sampled responses.

	model size	greedy decode	argmax	RAIL mean
STSB (RMSE↓)	XXS	1.078	1.448	**1.028**
	S	0.685	1.019	**0.649**
	L	0.628	0.989	**0.610**
			argmax	mean
STSB (AUC↑)	XXS	0.797	0.632	**0.889**
	S	0.895	0.820	**0.953**
	L	0.905	0.827	**0.961**
			argmax	median
Amazon reviews (MAE↓)	XXS	0.495	0.826	**0.474**
	S	0.301	0.444	**0.285**
	L	0.294	0.541	**0.291**

Table 2: Comparison of inference strategies on PaLM-2 models for different datasets and metrics. We draw 16 samples with an effective temperature of $T = \frac{1}{4}$ (via post-hoc scaling).

model	greedy	enumeration	sampling
FLAN-T5 S	4.419	2.407	**2.275**
FLAN-T5 L	0.455	0.410	**0.373**
FLAN-T5 XL	0.508	0.549	**0.457**

Table 3: Comparison of squared error (SE) on STSB with FLAN-T5 models. The sampling approach uses a temperature of 0.5.

4 Experiments and Discussion

We experimentally evaluate our proposed on NLP tasks with different evaluation metrics.

Datasets. We use two datasets. (i) Semantic Textual Similarity Benchmark (*STSB*) (Cer et al., 2017), which comprises of sentence pairs human-annotated with a similarity score from 0 to 5; since this is a regression task, we evaluate with the root mean squared error. (ii) *US Amazon reviews*, where we aim to predict the 5-star rating for a product review (Ni et al., 2019); since the task is in the form of ordinal regression, we use mean absolute error as the evaluation metric (Fathony et al., 2017). We list the prompts used in Table 6 (Appendix). In each case, we evaluate on samples of 1500 examples.

Models. We consider two instruction-tuned model families: PaLM-2 (Google and et al., 2023) and FLAN-T5 (Chung et al., 2022). We report results across different model sizes and temperatures. Unless otherwise stated, we fix the number of samples to $K = 16$, and the top-k parameter in decoding to 40 (Fan et al., 2018).

Methods. We evaluate the following methods: (i) greedy decoding, (ii) a baseline inspired from the self-consistency decoding of sampling K candidates and picking the one with the maximum likelihood (argmax) (Wang et al., 2023a), (iii) the proposed RAIL approach on the same K samples, and (iv) the temperature scaled variant of RAIL in §3.3 (denoted by a '*'). For (iv), we choose α so that the effective temperature is $\frac{1}{4}$.

Metric-aware inference helps. In Table 2, we report results across datasets and model sizes. We notice that RAIL improves over baselines across all model sizes on STSB and Amazon reviews (with the exception of model size *S*, where we see that median performs very similarly to the most likely generated sample).

Sampling versus enumeration. So far, when estimating the prediction maximizing the (2), we have used sampling from the LM distribution (see §3.2). Alternatively, if the targets are from a narrow interval (e.g., on the STSB dataset, the values are in the interval $[0, 5]$), one can score the model for targets enumerated at fixed intervals (e.g. $0, 0.5, 1.0, \ldots, 5.0$), and compute estimates for solutions in Table 1. In Table 3, we report results from FLAN-T5 on the STSB dataset for RAIL with both sampling and enumeration based estimates, where the latter is based on 11 equally spaced targets. We find that both sampling and enumeration lead to RAIL improving over choosing the most likely target. Further, we note that sampling is a more effective strategy than enumeration of equally spaced targets.

Role of model size. We find that the benefit from our technique reduces as the models increase in size. This sometimes coincides with a lowering entropy in predictions with increasing model size (see, e.g., results on Amazon in Table 7 in Appendix). We note this is consistent with prior works on MBR, which observed that as the model gets better, the optimal decision rule for EM (approximated by greedy decoding) performs comparable to the that for other metrics (Schluter et al., 2012). We stress that the gains we get with small and medium-sized models are still of large practical importance, especially in applications where deploying very large models is prohibitively expensive.

5 Conclusions

We have shown how regression and scoring-aware inference strategies can yield notable benefits for small and medium-sized LLMs. In the future, we wish to extend our approach to other less-explored evaluation metrics in the MBR literature; e.g., in Appendix B, we propose an F_1-score aware inference strategy and showcase its efficacy on TriviaQA (Joshi et al., 2017).

6 Limitations

There are multiple limitations of our work. First, we evaluate our proposed methods on multiple text datasets with numerical and text targets, however, many more types of outputs can be considered, including the time series targets. Next, it would be interesting to more systematically analyze how to efficiently solve the objective from (5) over many samples for text outputs for metrics like F_1 or BLEU, e.g. by means of dynamic programming. We also note that the datasets considered in this work are restricted to English. It would be interesting to expand the explorations to datasets in other languages.

7 Ethics Statement

All datasets used in this work are publicly available. No additional user data was collected or released as part of this work. All models used are publicly available and already pretrained, and no finetuning was conducted for any experiments. Instead, all experiments relied on running inference experiments with the models over several thousands of examples. Thus, the CO-2 footprint of this paper is minimal. We do not foresee any significant risks associated with this paper other than improving performance on tasks which are harmful.

References

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*.

Amanda Bertsch, Alex Xie, Graham Neubig, and Matthew Gormley. 2023. It's MBR all the way down: Modern generation techniques through the lens of minimum Bayes risk. In *Proceedings of the Big Picture Workshop*, pages 108–122, Singapore. Association for Computational Linguistics.

P.J. Bickel and K.A. Doksum. 1977. *Mathematical Statistics: Basic Ideas and Selected Topics*. Holden-Day series in probability and statistics. Holden-Day.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics.

Julius Cheng and Andreas Vlachos. 2023. Faster minimum bayes risk decoding with confidence-based pruning. *arXiv preprint arXiv:2311.14919*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Nicola Ehling, Richard Zens, and Hermann Ney. 2007. Minimum bayes risk decoding for bleu. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 101–104.

Bryan Eikema and Wilker Aziz. 2020. Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation.

Rizal Fathony, Mohammad Ali Bashiri, and Brian Ziebart. 2017. Adversarial surrogate losses for ordinal regression. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Mara Finkelstein and Markus Freitag. 2024. MBR and QE finetuning: Training-time distillation of the best and most expensive decoding methods. In *The Twelfth International Conference on Learning Representations*.

Markus Freitag, Behrooz Ghorbani, and Patrick Fernandes. 2023. Epsilon sampling rocks: Investigating sampling strategies for minimum Bayes risk decoding for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9198–

9209, Singapore. Association for Computational Linguistics.

Gemini Team and Rohan Anil et al. 2023. Gemini: A family of highly capable multimodal models.

Tilmann Gneiting and Adrian E Raftery. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378.

Google and Rohan Anil et al. 2023. Palm 2 technical report.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. 2023. Large language models are zero-shot time series forecasters.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 169–176, Boston, Massachusetts, USA. Association for Computational Linguistics.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. 2023. Large language models with controllable working memory. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1774–1793, Toronto, Canada. Association for Computational Linguistics.

Frederick Liu, Terry Huang, Shihang Lyu, Siamak Shakeri, Hongkun Yu, and Jing Li. 2022. Enct5: A framework for fine-tuning t5 as non-autoregressive models.

Tiedong Liu and Bryan Kian Hsiang Low. 2023. Goat: Fine-tuned llama outperforms gpt-4 on arithmetic tasks.

Ali Naseh, Kalpesh Krishna, Mohit Iyyer, and Amir Houmansadr. 2023. Stealing the decoding algorithms of language models. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, CCS '23, page 1835–1849, New York, NY, USA. Association for Computing Machinery.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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 188–197, Hong Kong, China. Association for Computational Linguistics.

OpenAI et al. 2023. Gpt-4 technical report.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2023. Large language models are effective text rankers with pairwise ranking prompting.

Ralf Schluter, Markus Nussbaum-Thom, and Hermann Ney. 2012. Does the cost function matter in bayes decision rule? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):292–301.

Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. 2022. Follow the wisdom of the crowd: Effective text generation via minimum bayes risk decoding. *arXiv preprint arXiv:2211.07634*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Kazuki Uematsu and Yoonkyung Lee. 2015. Statistical optimality in multipartite ranking and ordinal regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(5):1080–1094.

Robert Vacareanu, Vlad-Andrei Negru, Vasile Suciu, and Mihai Surdeanu. 2024. From words to numbers: Your large language model is secretly a capable regressor when given in-context examples. *arXiv preprint arXiv:2404.07544*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023a. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Yihan Wang, Si Si, Daliang Li, Michal Lukasik, Felix Yu, Cho-Jui Hsieh, Inderjit S Dhillon, and Sanjiv Kumar. 2023b. Two-stage llm fine-tuning with less specialization and more generalization.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jianhao Yan, Jin Xu, Fandong Meng, Jie Zhou, and Yue Zhang. 2022. Dc-mbr: Distributional cooling for minimum bayesian risk decoding. *arXiv preprint arXiv:2212.04205*.

Zhen Yang, Ming Ding, Qingsong Lv, Zhihuan Jiang, Zehai He, Yuyi Guo, Jinfeng Bai, and Jie Tang. 2023. Gpt can solve mathematical problems without a calculator.

A Further related work

Minimum Bayes risk decoding. As noted in the introduction, prior work on MBR have considered optimizing for common metrics in the machine translation and text generation literature. The closest to our paper is the work of (Wang et al., 2023a), who considered sampling from the model distribution when applied with chain of thought prompting, and showed how majority vote improves over the baseline under different arithmetic and reasoning tasks. Other works explored different aspects of MBR, including: the role of the sampling algorithms (Freitag et al., 2023; Cheng and Vlachos, 2023), how label smoothing interacts with MBR (Yan et al., 2022), and how it generalizes other techniques (Suzgun et al., 2022; Bertsch et al., 2023). (Finkelstein and Freitag, 2024) recently considered distillation of MBR solution from the teacher to a student model so as to avoid the overhead induced by MBR at inference time.

Finetuning approaches for target task alignment. Previous works considered approaches for aligning the models for target datasets. For example, soft prompts were finetuning on target datasets without loosing generalization to other tasks (Wang et al., 2023b), and general finetuning was conducted on carefully tailored datasets for improved model robustness (Li et al., 2023). In our work, we focus on zero-shot setting where no fine-tuning is conducted.

Finetuning approaches for numerical tasks. Autoregressive finetuning of LLMs on numerical tasks with CoT has been found effective (Liu and Low, 2023). One line of work for modeling predictive tasks with pre-trained Transformer based models is to add a regression head on top of the transformed/pooled encoded input tokens and finetune the resulting model on numerical targets using a regression loss. This is an approach which has been for encoder based models (e.g. Bert), and has also been applied to encoder-decoder (e.g. T5) models (Liu et al., 2022), and these approaches could be extended to decoder models too. In a similar line of work, an embedding can be extracted from a decoder model finetuned on modified attention mask and additional tasks (BehnamGhader et al., 2024). In this work, we focus on the zero shot approaches, and so we leave training approaches for future work.

B Additional results on F_1 maximization on Trivia QA

We extend our approach to the F_1 score evaluation metric. Consider a reading comprehen-

	model size	greedy decode	T=0.25 argmax	T=0.25 mean	T=0.25 w-mean	T=0.5 argmax	T=0.5 mean	T=0.5 w-mean	T=1.0 argmax	T=1.0 mean	T=1.0 w-mean
STSB	XXS	1.078	1.126	1.043	1.028	1.241	1.021	0.992	1.448	1.007	**0.978**
	S	0.685	0.787	0.643	0.649	0.908	**0.636**	0.642	1.019	0.641	0.641
	L	0.628	0.729	0.592	0.610	0.852	0.582	0.586	0.989	**0.580**	**0.580**
			T=0.25 argmax	T=0.25 median	T=0.25 w-median	T=0.5 argmax	T=0.5 median	T=0.5 w-median	T=1.0 argmax	T=1.0 median	T=1.0 w-median
Amazon reviews	XXS	0.495	0.509	0.484	**0.474**	0.624	0.485	0.487	0.826	0.493	0.493
	S	0.301	0.290	0.297	**0.285**	0.329	0.300	0.297	0.444	0.299	0.299
	L	0.294	0.318	0.293	**0.291**	0.380	0.294	0.293	0.541	0.298	0.295
			T=0.25 argmax	T=0.25 F_1	T=0.25 w-F_1	T=0.5 argmax	T=0.5 F_1	T=0.5 w-F_1	T=1.0 argmax	T=1.0 F_1	T=1.0 w-F_1
Trivia-QA	XXS	0.314	0.300	0.319	0.318	0.255	0.323	**0.326**	0.178	0.307	0.304
	S	0.620	0.656	0.626	**0.678**	0.658	0.641	0.662	0.636	0.650	0.650
	L	0.886	**0.888**	0.886	**0.888**	**0.888**	0.883	0.887	0.887	0.880	0.885

Table 4: Root mean squared error (RMSE) on STSB dataset (the lower the better), Mean absolute error (MAE) on Amazon reviews dataset (the lower the better), and F_1 metrics on Trivia-QA dataset (the higher the better) from PaLM-2 models of varying size. We report different methods of inference across different temperatures. For the weighted approaches, we fix the sampling temperature to $T = 1$ and accordingly vary the α in (5) so as to arrive at the effective temperature equal to the value reported.

model	w/ pairs	w/o pairs
PaLM-2 XXS	0.302	0.295
PaLM-2 XS	0.678	0.670
PaLM-2 L	0.886	0.887

Table 5: Performance of RAIL (as evaluated by F_1) on TriviaQA with and without the inclusion of concatenated pairs in the candidate set.

sion task, where the F_1 score is the evaluation metric $m(y, \hat{y})$, defined by the harmonic mean of recall$(y, \hat{y}) = \frac{|y \cap \hat{y}|}{|y|}$ and precision$(y, \hat{y}) = \frac{|y \cap \hat{y}|}{|\hat{y}|}$. To illustrate the task, suppose for the question "`What is the hottest month in the year`", the responses and associated probability from an LM are {"`July`": 0.25, "`July 2023`": 0.23, "`Month of July`": 0.24, "`May`": 0.28}. The mode of this distribution is "`May`"; whereas the maximizer of (2) is "`July`".

To optimize the F_1 metric, we solve (7) over a candidate set C, which we choose to contain the K samples and additional targets derived from them.

$$\hat{y}(x) \in \operatorname*{argmax}_{y' \in C} \sum_{i=1}^{K} m(y_i, y'). \tag{7}$$

While the F_1 score does not admit a closed-form solution, as is the case for the metrics listed in Table 1, we make an observation that its formulation allows for introducing a different form of efficiency. In particular, we notice that due to the trade-off between precision and recall in the F_1 score formulation, the following candidate set construction can lead to increasing recall at the expense of precision, thus providing a way to cheaply enumerate additional reasonable candidates.

Candidate set construction. One simple choice for the candidate set C could be take the K sampled outputs, i.e., $C = \{y_1, \ldots, y_K\}$. One may additionally include in this set transformations on each y_i or new candidates formed from combining two or more of the samples.

For reading comprehension or question-answering applications, where the output is a list of keywords that constitute an answer to a question, one may additionally include samples formed by concatenating pairs of sampled outputs, i.e., $\texttt{concat}(y_i, \texttt{delim}, y_j), \forall i \neq j$. These concatenated answers have the effect of increasing recall, at the cost of lower precision. We follow that procedure for the Trivia-QA experiments.

In Table 4, we provide results on Trivia-QA reading comprehension task (Joshi et al., 2017) with the proposed F_1-aware inference strategy.

To additionally analyze the effectiveness of the candidate set augmentation, in Table 5 we compare the performance of RAIL (specifically the temperature scaled variant) with and without the inclusion of concatenated pairs in the candidate set. For both the XXS and S models, the inclusion of concatenated pairs is seen to yield a significant improvement in F_1-score.

Dataset	Prompt
STSB	What is the sentence similarity between the following two sentences measured on a scale of 0 to 5: {Sentence #1}, {Sentence #2}. The similarity measured on a scale of 0 to 5 with 0 being unrelated and 5 being related is equal to
Amazon reviews	What is the rating corresponding to the following review in the scale of 1 to 5, where 1 means negative, and 5 means positive? Only give a number from 1 to 5 with no text. Review: {Review} Rating:
Trivia-QA	Answer the following question without any additional text. Question: {Question}. Answer:

Table 6: Prompts used for different datasets. Curly braces denote inputs specific to an input example.

C Additional details

In Table 6 we report the prompts we used in our experiments for zero-shot inference.

For all datasets, we use validation splits, and where not available, we use the first 1500 examples from the train split.

The datasets are publicly available, for example from the `tensorflow.org` platform:

- `https://www.tensorflow.org/datasets/catalog/glue#gluestsb`,
- `https://www.tensorflow.org/datasets/catalog/amazon_us_reviews`,
- `https://www.tensorflow.org/datasets/catalog/trivia_qa`.

model	STSB	Amazon	Trivia-QA
PaLM-2 XXS	1.141	1.064	1.328
PaLM-2 XS	1.055	0.753	0.475
PaLM-2 L	0.976	0.361	0.186

Table 7: Empirical entropy across model sizes and datasets.

samples	XXS	S	L
(Greedy Decode)	1.078	0.685	0.628
2	1.044	0.679	0.624
4	1.036	0.669	0.613
6	1.031	0.664	0.607
8	1.028	0.660	0.603
10	1.025	0.657	0.601
12	1.024	0.655	0.600
14	1.022	0.653	0.599
16	1.021	0.652	0.598

Table 8: RMSE as a function of the number of samples on STSB across PaLM-2 models of varying size. Results for temperature $T = 0.25$.

D Additional experiments

In Table 7 we report empirical entropy estimates as measured based on the 16 samples generated from the model. We find that entropy decreases as model size increases. We observe a particularly sharp decrease in entropy for the Amazon reviews and Trivia-QA datasets, where for larger model sizes we don't find improvements from RAIL approaches.

In Table 4 we report RMSE on STSB dataset, MAE on Amazon reviews dataset, and F_1 metrics on Trivia-QA dataset from PaLM-2 models of varying size across multiple temperature values. We find improvements over baselines on STSB and Amazon reviews datasets for most temperatures. For Trivia-QA, we find improvements for XXS and S models for some temperatures, and for L, we don't find a difference from our methods due to low entropy in the responses (see Table 7). In Table 10 we additionally report Pearson correlation metrics on STSB, confirming the results of RAIL improving over autoregressive inference. Lastly, in Table 9 we report cost weighted multi-class AUC with costs corresponding to the difference between the annotated labels: $|y_1 - y_2|$. We find on both STSB and Amazon reviews datasets that the optimal decision rule (mean over the distribution) improves over the baselines.

In Table 8, we report the impact of the number of samples on the results. We note that there is an improvement in the results with the increase in the number of samples, however beyond 8 samples there is a diminishing improvement in practice. On STSB with temperature $\frac{1}{4}$, even with as few as *two* samples, our method starts to show improvements over greedy decoding.

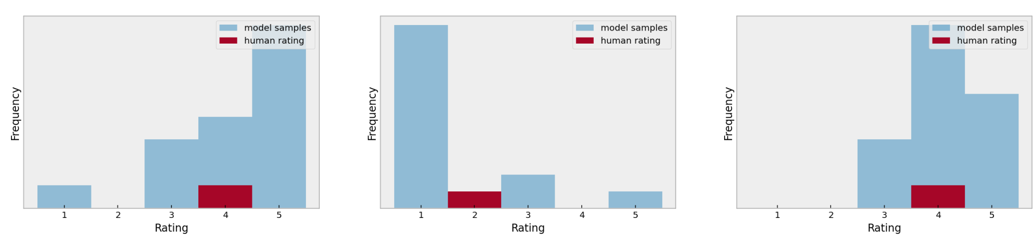
In Figure 2 we report examples from the Amazon dataset and the corresponding: human annotations and samples from the model. Notice how samples cover significant proportions of the ratings. We find that the samples end up in the vicinity of the human annotation, and thus in many cases taking a *mean* over samples helps improve the prediction over the *mode*.

	model size	greedy decode	T=0.25		T=0.5		T=1.0	
			argmax	mean	argmax	mean	argmax	mean
STSB	XXS	0.797	0.755	0.882	0.714	0.890	0.632	0.889
	XS	0.895	0.870	0.950	0.843	0.954	0.820	0.953
	L	0.905	0.885	0.948	0.859	0.959	0.827	0.961
Amazon reviews	XXS	0.87	0.894	0.925	0.866	0.94	0.788	0.942
	XS	0.9	0.91	0.925	0.914	0.941	0.9	0.958
	L	0.925	0.922	0.951	0.906	0.962	0.837	0.964

Table 9: Cost-weighted multi-partite AUC metrics on STSB and Amazon datasets (*the higher the better*). RAIL methods improve over the baselines. See §3.4 for the definition of AUC we use. We assume costs to correspond to the difference between the annotated labels: $|y_1 - y_2|$.

model	greedy decode	T=0.25		T=0.5		T=1.0	
		argmax	mean	argmax	mean	argmax	mean
PaLM-2 XXS	0.767	0.738	**0.790**	0.670	**0.790**	0.544	0.786
PaLM-2 XS	0.898	0.878	**0.915**	0.852	0.913	0.821	0.910
PaLM-2 L	0.909	0.893	0.920	0.881	0.922	0.860	**0.923**

Table 10: Pearson correlation metrics on STSB. RAIL methods improve over the baselines.



(a) *It is a nice color of black and my husband likes how it feels in his hand.*

(b) *This item is a good idea. However, Unless the ear canal is reasonably deep (...) it's of no use. The plastic hooks that come with it are hard and too small (...). Might be good for children.*

(c) *One of the sides is made for apple products, the other is just standard usb. Both will work with apple products, just one side (the A side) charges faster. Other than that, it's fantastic. :D*

Figure 2: Examples from the Amazon dataset and the corresponding: human annotations and samples from the model. We find that in many cases, taking into account the model distribution (i.e. a *mean* of the distribution) allows for a prediction closer to the annotation than simply taking the *mode* of the distribution.