WHEN CAN YOU TRUST LARGE LANGUAGE MODELS?

Radu Paradovschi, Darvin Yi, Andrew Rabinovich, Zhao Chen
Upwork Inc.
Palo Alto, CA 94301, USA
{alexradu, darvinyi, andrewrabinovich, zhaochen}@upwork.com

ABSTRACT

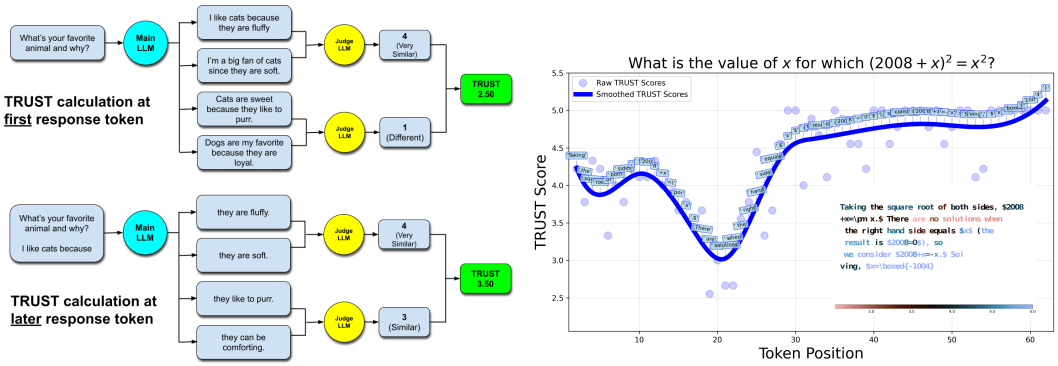
Quantifying neural network model uncertainty is a difficult problem that has far-reaching implications on our ability to improve model reliability. Unfortunately, uncertainty quantification is especially difficult in the context of LLMs, as standard methods for uncertainty measurement either rely on single-token outputs or have other structural assumptions that limit utility in practical settings. We will show that these methods do not capture uncertainty well in challenging environments such as difficulty quantification and hallucination detection, and introduce TRUST (Temperature-Related Unambiguity via Similarity Tracking) scores as a way to easily generalize uncertainty methods to all text-in, text-out settings. TRUST scores calculate uncertainty based on semantic similarity of multiple output rollouts for an LLM model, can be calculated without any white-box access to model internals, and strongly outperforms standard methods in quantifying LLM uncertainty.

1 INTRODUCTION

One known limitation of modern AI models is that they do not know what they do not know. Especially as LLMs begin to be adopted by safety-critical industries like finance (Easin et al., 2024) and healthcare (Singhal et al., 2025), understanding LLM uncertainty will become increasingly critical to ensure AI continues to behave robustly. Current proposed solutions rely on statistical measures like entropy (Shannon, 1948) and max softmax probability (MSP) (Hendrycks & Gimpel, 2016; Pearce et al., 2021), which are largely insufficient for uncertainty estimation within *autoregressive* models as semantic meaning of LLM outputs generally spans the entire output. More sophisticated methods like semantic entropy (Kuhn et al., 2023) show promising initial abilities to reason about uncertainty on semantic multi-token outputs, but are often impractical in realistic settings with longer and more complex outputs. In general, previous uncertainty work suffers from scope restrictions due to fixating on accuracy metrics in fact-retrieval settings as opposed to true uncertainty benchmarks.

Although newer methods like semantic entropy can work in simple settings (Appendix A.4), their performance surprisingly collapses when tested in less restricted environments like quantifying difficulty (Section 4.3) and long-output hallucination detection (Section 4.4). To recover performance, we hypothesize that we need to produce a method that can work in general text-in text-out autoregressive settings without any additional assumptions on the model or input/output structure.

We propose TRUST scores as a potential solution for general LLM uncertainty prediction. TRUST scores are computed by sampling multiple candidate outputs from an input, and then using a separate LLM judge model to compute an average pairwise semantic similarity between the IID sampled responses. In this way, TRUST scores are agnostic to response, length, input length, token sampling strategy, and to model architecture. Crucially, calculating TRUST scores does not require any white-box access to model internals. We will demonstrate that TRUST scores are on par with semantic entropy in simple settings, and greatly outperform all baselines in more semantically complex settings. TRUST can also be distilled into a more efficient inference model for more practicality. We therefore see TRUST scores as a first uncertainty prediction method that is applicable to all LLM settings with no additional assumptions or dataset processing.



(a) Calculation of TRUST scores

(b) Visualization of TRUST scores in the MATH dataset

Figure 1: Diagram of calculation of TRUST scores (a) both at the first token and at a later token using a similarity scale from 1 to 5, with 5 being most similar. Visualization of TRUST scores (b) on entries in the MATH dataset. In the colorized example, tokens that are colored red correspond to lower TRUST scores and higher uncertainty.

2 Related Work

LLM uncertainty commonly uses standard white-box statistical methods like entropy (Wang et al., 2022) and max softmax (MSP) (Liu et al., 2023). Semantic uncertainty (Kuhn et al., 2023) is a recent method that extends uncertainty estimation to multiple rollouts but limits itself to fact-retrieval benchmarks and shorter outputs. Uncertainty detection in LLMs is also often packaged with hallucination detection (e.g. (Rawte et al., 2023)), but such methods but such methods generally rely on references to external sources of truth. LLMs can express their uncertainty directly (Lin et al., 2022), but these methods tend to lead to false signals and overconfidence (Xiong et al., 2023). Uncertainty estimation is also tightly related to long-tail detection (Lakshminarayanan et al., 2017; Vyas et al., 2018) and can be directly used to steer training (Yang & Loog, 2016; Shi et al., 2020).

3 Methods

3.1 Uncertainty for Complex LLM Outputs

Restrictions of uncertainty estimation techniques to next-token prediction like in entropy or MSP are clearly insufficient in natural language where meaning can span the full output. One notable attempt to extend entropy to multi-token semantic outputs is Semantic Entropy (Kuhn et al., 2023), which has shown to be effective at predicting model accuracy within a variety of fact retrieval settings like TriviaQA (Joshi et al., 2017) and CoQA (Reddy et al., 2019). However, from a practicality perspective, semantic entropy requires multiple entailment calculations from a custom entailment model to group potential outputs into equivalence classes. But more importantly, its focus on concrete sentences as the main input unit for uncertainty measurement does not generalize well beyond output text that is decomposable into discrete factoids, as we will demonstrate empirically later.

Another fundamental weakness of many prior uncertainty works like (Kuhn et al., 2023) is that they focus on correlations between uncertainty and *accuracy*. We presume that part of this design decision lies in practicality, as accuracy benchmarks are much more abundant than true uncertainty benchmarks. But we feel it is crucial to decouple accuracy from confidence, and to focus on the latter in treatments of uncertainty estimation. We will show that more challenging benchmarks that directly probe uncertainty like problem difficulty or hallucination rates will cause methods like semantic entropy to underperform.

TRUST is designed to be general within all text-in text-out settings, lacks any structural assumptions, and is tested against pure uncertainty benchmarks rather than accuracy benchmarks, to avoid signal dilution when models are e.g. confidently incorrect. TRUST therefore rectifies both generalization issues with previous methods as well as limitations within prior evaluation settings.

Criteria	Compatibility matrix for uncertainty methods				
	$\widehat{\text{TRUST}}$	TRUST	SE	MSP	Entropy
Unstructured long-form outputs	✓	✓	×	∼	∼
Multi-token	✓	✓	✓	×	×
Black-box	✓	✓	✓	×	×

Table 1: Compatibility matrix for different uncertainty detection methods across multiple criteria. Methods which are compatible with "Unstructured long-form outputs" do not place additional structure on inputs such as how Semantic Entropy induces factoid decomposition on longer text. Although MSP and Entropy technically do not assume any structure in output, they are generally ill-equipped to deal with the semantic complexities of such situations due to their lack of multi-token context windows, as we will empirically demonstrate in Section 4. Multi-token methods model the joint distribution of complete rollouts. Black-box methods do not require access to predicted log-probabilities.

A compatibility matrix for different uncertainty methods across several dimensions is presented in Table 1.

3.2 Computing TRUST Scores

We take advantage of the temperature-induced output variance of a model $\mathcal{M}$. Denote the input query of the model as $\mathbf{q}^{(j)}$ and a partial completion of t tokens (t might be 0) as $\mathbf{r}^{(j,t)}$. The complete prefix input into a language model is the concatenation $\mathbf{x}^{(j)} = \text{Concat}(\mathbf{q}^{(j)}, \mathbf{r}^{(j,t)})$. We then generate $2N$ completions at temperature $\tau > 0$, usually until a stop token is encountered. Denote these completions $\mathbf{y}_i^{(j)}$, $i \in (1, \ldots, 2N)$. We then initialize a judge model J, usually a pre-trained LLM, which produces semantic similarity scores when comparing two pieces of text. The TRUST score at token index t for example j is defined as

$$\text{TRUST}_{j,t} = E_{i=1,\ldots,N}\left[J\left(\text{Concat}(\mathbf{r}^{(j,t)}, \mathbf{y}_{2i-1}^{(j)}), \text{Concat}(\mathbf{r}^{(j,t)}, \mathbf{y}_{2i}^{(j)})\right)\right] \tag{1}$$

The TRUST computation is schematically illustrated in Figure 1a. To save compute, generally we will only generate $N + 1$ completions and make pairwise comparisons between i and $i + 1$ in the above expectation rather than between $2i - 1$ and $2i$.

TRUST is effective as a measure of total uncertainty, without any explicit decomposition into epistemic or aleatoric uncertainty. Further discussion of this point can be found in Appendix A.1. We also present theory that TRUST is related to MSP scores in appropriate limits in Appendix A.2.

3.3 TRUST Modeling

Although Section 3.2 allows us to generate TRUST scores, generating N trial completions at each token position to produce TRUST scores can be expensive and time-consuming. Thus, we also test performance of a model trained on TRUST scores. These can be simple language predictive models, and we will use a BERT (Devlin et al., 2019) model within this work. We will show that a predictive model trained on TRUST scores is competitive with using the raw TRUST scores themselves, demonstrating that we can avoid the computational downsides of having to compute TRUST scores at inference time.

4 Experiments

4.1 Baseline Methods

Our baseline measurements for uncertainty consist of Shannon entropy ($-\sum p_i \log(p_i)$), Maximum Softmax Probability (MSP), expressed uncertainty (LLMs are asked to directly output uncertainty), and Semantic Entropy (Kuhn et al., 2023). We also compare against ensembling uncertainty in Section 4.3. For more details, please refer to Appendix A.10. In all following sections, we also have

LLM	MATH Difficulty Mean Squared Error (MSE) ↓					
	$\widehat{\text{TRUST}}$	TRUST	SE	EU	MSP	Entropy
GPT	**1.18 ± 0.04**	**1.20 ± 0.04**	1.45 ± 0.01	1.52 ± 0.26	1.47 ± 0.04	1.46 ± 0.04
Llama	**1.22 ± 0.05**	**1.22 ± 0.02**	1.44 ± 0.05	1.37 ± 0.05	1.42 ± 0.03	1.58 ± 0.13

Table 2: MSE for the difficulty prediction task in the MATH dataset. GPT is gpt-4o-mini and Llama is llama-3.1-70b. SE is Semantic Entropy and EU is Expressed Uncertainty.

a $\widehat{\text{TRUST}}$ method which is a distilled model trained on TRUST scores as a target, to show that these scores are learnable for faster inference. The distilled model architecture is shown in Appendix A.9.

4.2 Simple Uncertainty Prediction

As a simple benchmark, we create a toy dataset of "conflicting opinions" - the inputs are questions that admit multiple answers (e.g. "What is your favorite animal?") and outputs have various levels of mixture of different responses. We test if various uncertainty methods can successfully predict the mixing ratio as a proxy for uncertainty. Details of this implementation and results are discussed in Appendix A.4. TRUST and Semantic Entropy both outperform classical methods like MSP and entropy, which is expected in this simple setting. We now proceed to more complex uncertainty settings.

4.3 Difficulty Prediction on the MATH Dataset

The MATH dataset (Hendrycks et al., 2021) consists of open-ended math problem labeled with difficulties from 1 to 5, which we use as a proxy for model uncertainty. A good uncertainty measure should be able to match an input problem with a difficulty level.

Results are displayed in Table 2 for multiple LLM models (GPT-4o-mini and llama3.1-70b) generating the response. TRUST scores outperform other methods significantly, and this outperformance is even visible by eye (Figure 1b) where TRUST clearly separates between the different levels. Not shown is the ensembled uncertainty Fadeeva et al. (2023), as it is only applicable to open-source models like Llama, but we did run the Llama ensemble benchmark and produced the result 1.43 ± 0.05. Semantic Entropy underperforms here even though its performance is reasonable in our toy setting, which we attribute to the complexity of MATH outputs and to Semantic Entropy's structural assumptions (reliance on entailment and simple factoids) to capture the correct long-form semantics in this dataset.

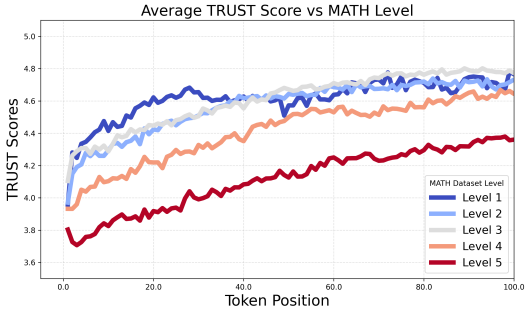
4.4 Hallucination detection

We created a dataset for hallucination detection which contains a total number of 2100 examples sampled equally from MedHallu (Pandit et al., 2025), HaluEval (Li et al., 2023); FactCHD (Chen et al., 2023), and RAGTruth (Niu et al., 2024). We regenerated the responses and the hallucination labels at the token level (details in Appendix A.8). This resulted in a larger dataset that allows us to check for hallucinations in dynamically generated text by an LLM, rather than relying on set tokens in a pre-existing dataset. Hallucination labels range from factually grounded responses (level 1) to entirely fabricated responses (level 5). We plan to make the full dataset available at a later time.

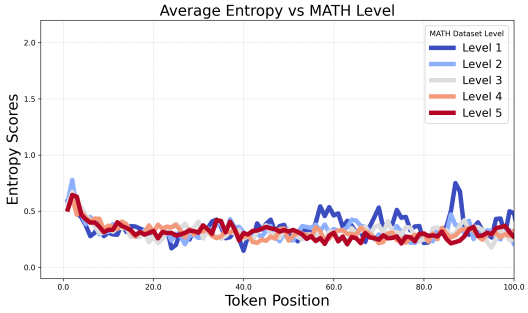
As shown in Table 3 TRUST significantly outperforms all other baselines in terms of MSE and AUROC when fitting logistic regression onto token-level hallucination labels. As in the MATH experiments, we see in Figure 2(e) that TRUST separates different hallucination levels clearly.

5 Conclusion

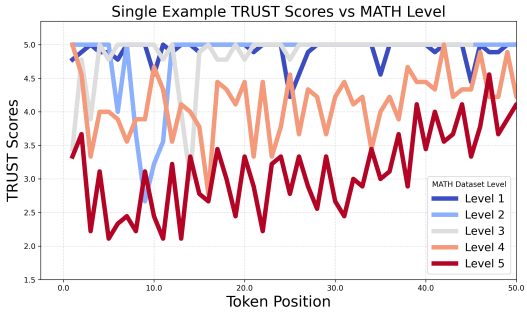
Uncertainty estimation remains a surprisingly difficult problem for current methods, from classic algorithms like entropy and MSP to more modern work like semantic entropy. We showed that these methods work poorly in real settings with longer input/outputs, and hypothesize that methods thus far impose too many structural assumptions (single-token, factual outputs, etc.) to their problem setting.



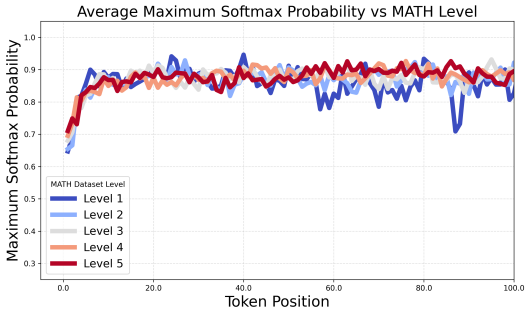
(a) TRUST (averaged) on MATH.



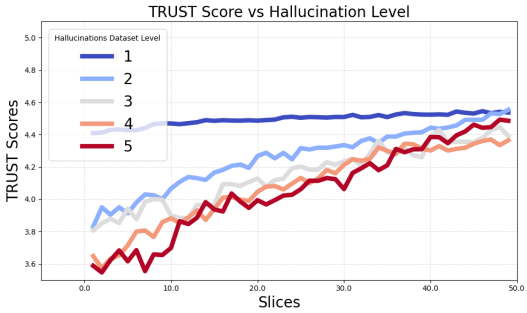
(b) Entropy on MATH.



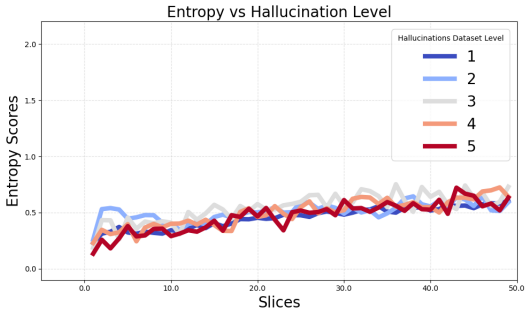
(c) TRUST (single) on MATH.



(d) MSP on MATH.



(e) TRUST (averaged) on hallucinations.



(f) Entropy on hallucinations.

Figure 2: Comparing TRUST uncertainty averaged across the dataset (a) and for a few single examples (c) at different token positions for MATH difficulty level versus Entropy (b) and MSP (d). Comparing TRUST uncertainty (e) versus Entropy (f) for hallucination detection. TRUST (a, c, d) scores exhibit clear separation between levels, while other metrics (b, d, f) do not.

		Hallucination Level Prediction				
LLM	Metric	$\widehat{\text{TRUST}}$	TRUST	SE	MSP	Entropy
GPT	AUROC ↑	**0.60 ± 0.03**	**0.75 ± 0.01**	0.51 ± 0.01	0.50 ± 0.01	0.50 ± 0.01
GPT	MSE ↓	**0.36 ± 0.14**	**0.24 ± 0.02**	0.60 ± 0.10	0.57 ± 0.10	0.78 ± 0.01

Table 3: AUROC and MSE for the hallucinations prediction task. SE is Semantic Entropy.

To remedy this, we introduced TRUST scores, a novel method that enables semantic reasoning across the entire LLM output while not requiring any structural assumptions or special access to model internals. We showed that TRUST scores perform especially well within more complex settings, which sets it apart from pre-existing methods. We hope TRUST will make uncertainty estimation much more accessible and accurate for LLM applications, which is especially critical if we aim to keep LLMs safe as they permeate more and more aspects of our digital lives.

IMPACT STATEMENT

This work presents a new method to detect uncertainty within LLMs, which may be used to inform users when LLMs are producing poor outputs or hallucinating. We believe this type of work will have positive broader impacts, especially as LLMs become more widespread and robust methods to quantify their trustworthiness become more critical to AI safety.

REPRODUCIBILITY STATEMENT

The authors of this work were careful to detail all processes and assumptions and ensure that the results within this manuscript are reproducible. All datasets used are either public or in the case of the experiments within a controlled setting (Section A.4) will be made available at a later time. All implementation details necessary to reproduce our work are provided in Section 4 for specific experimental settings, and Section 3 for TRUST score computation. Appendix A.5 and A.7 provides all LLM prompts used in our work, and Appendix A.9 provides all architectural details on how to train a distilled $\widehat{\text{TRUST}}$ model. Details in Appendix A.10 provide exact equations for computations of baselines.

REFERENCES

Xiang Chen, Duanzheng Song, Honghao Gui, Chenxi Wang, Ningyu Zhang, Yong Jiang, Fei Huang, Chengfei Lv, Dan Zhang, and Huajun Chen. Factchd: Benchmarking fact-conflicting hallucination detection. *arXiv preprint arXiv:2310.12086*, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Arafat Md Easin, Saha Sourav, and Orosz Tamás. An intelligent llm-powered personalized assistant for digital banking using langgraph and chain of thoughts. In *2024 IEEE 22nd Jubilee International Symposium on Intelligent Systems and Informatics (SISY)*, pp. 625–630. IEEE, 2024.

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, et al. Lm-polygraph: Uncertainty estimation for language models. *arXiv preprint arXiv:2311.07383*, 2023.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *ICLR*, 2021.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*, 2023.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. Halueval: A large-scale hallucination evaluation benchmark for large language models. *arXiv preprint arXiv:2305.11747*, 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*, 2022.

Bo Liu, Liming Zhan, Zexin Lu, Yujie Feng, Lei Xue, and Xiao-Ming Wu. How good are llms at out-of-distribution detection? *arXiv preprint arXiv:2308.10261*, 2023.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*, 2023.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Randy Zhong, Juntong Song, and Tong Zhang. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10862–10878, 2024.

Shrey Pandit, Jiawei Xu, Junyuan Hong, Zhangyang Wang, Tianlong Chen, Kaidi Xu, and Ying Ding. Medhallu: A comprehensive benchmark for detecting medical hallucinations in large language models. *arXiv preprint arXiv:2502.14302*, 2025.

Tim Pearce, Alexandra Brintrup, and Jun Zhu. Understanding softmax confidence and uncertainty. *arXiv preprint arXiv:2106.04972*, 2021.

Vipula Rawte, Amit Sheth, and Amitava Das. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*, 2023.

Siva Reddy, Danqi Chen, and Christopher D Manning. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019.

Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27 (3):379–423, 1948.

Weishi Shi, Xujiang Zhao, Feng Chen, and Qi Yu. Multifaceted uncertainty estimation for label-efficient deep learning. *Advances in neural information processing systems*, 33:17247–17257, 2020.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, et al. Toward expert-level medical question answering with large language models. *Nature Medicine*, 31(3):943–950, 2025.

Freddie Bickford Smith, Jannik Kossen, Eleanor Trollope, Mark van der Wilk, Adam Foster, and Tom Rainforth. Rethinking aleatoric and epistemic uncertainty. *arXiv preprint arXiv:2412.20892*, 2024.

Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 550–564, 2018.

Haoyu Wang, Hongming Zhang, Yuqian Deng, Jacob R Gardner, Dan Roth, and Muhao Chen. Extracting or guessing? improving faithfulness of event temporal relation extraction. *arXiv preprint arXiv:2210.04992*, 2022.

Lisa Wimmer, Yusuf Sale, Paul Hofman, Bernd Bischl, and Eyke Hüllermeier. Quantifying aleatoric and epistemic uncertainty in machine learning: Are conditional entropy and mutual information appropriate measures? In *Uncertainty in artificial intelligence*, pp. 2282–2292. PMLR, 2023.

Yijun Xiao and William Yang Wang. On hallucination and predictive uncertainty in conditional language generation. *arXiv preprint arXiv:2103.15025*, 2021.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*, 2023.

Yasin Abbasi Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvári. To believe or not to believe your llm. *arXiv preprint arXiv:2406.02543*, 2024.

Yazhou Yang and Marco Loog. Active learning using uncertainty information. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 2646–2651. IEEE, 2016.

Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 93–102, 2019.

A APPENDIX

A.1 DISCUSSION

One common discussion within the AI uncertainty literature, including for LLM uncertainty (Yadkori et al., 2024), is whether proposed methods are more sensitive to epistemic or aleatoric uncertainty. We position TRUST as an estimate of *total uncertainty*, especially as the classic decomposition of uncertainty into its aleatoric and epistemic components can be ill-defined within discrete prediction spaces like language (Wimmer et al., 2023; Smith et al., 2024). We note that the experimental setting explored in Appendix A.4 seems like a classic example of aleatoric uncertainty (i.e. *single input* $\mapsto$ *multiple valid outputs*). In contrast, the MATH dataset setting in Section 4.3 is a good example of epistemic uncertainty measurement, as the data points within the dataset are all well-defined, solvable math problems that are in-principle learnable by a model. And hallucinations are often attributed to epistemic uncertainty (Xiao & Wang, 2021) but can also arise from aleatoric dataset variance.

In the future, we will explore further decompositions of TRUST scores into model and text prefix components, following the classic Bayesian decomposition $P(\text{out}|\text{in}) = P(\text{out}, \text{in})/P(\text{in})$. These studies will extend TRUST to capture uncertainties that are invariant to the generating LLM, while in this current work we always generate TRUST with respect to a specific data source and model.

We reiterate that TRUST scores are sensitive to model uncertainty, *without consideration of whether the model is accurate*. Accuracy and uncertainty have often been tested together (Manakul et al., 2023; Kuhn et al., 2023), and loss prediction has been used as a proxy for uncertainty detection (Yoo & Kweon, 2019), but a direct measurement of model uncertainty provides more confidence. However, we do note that our hallucination detection experiments involve an implicit reference ground truth, and TRUST outperformed other methods in that setting as well.

TRUST scores are a total uncertainty metric tailored for the LLM era that extends well-motivated statistical methods like MSP to multi-token outputs by leveraging the semantic understanding of LLMs, while requiring zero access to model internals. Methods like TRUST represent a "best of both worlds" approach, where LLM semantic analysis augments statistically grounded metrics to produce uncertainty measurements that are theoretically sound while being effective in realistic natural language settings.

A.2 THEORY

We can show that TRUST scores for single-token prediction and under mild conditions are related to the squared maximum softmax probability (MSP) (Hendrycks & Gimpel, 2016), a measure of uncertainty that has been widely used in industry since its invention.

Theorem 1. *Take a sequence prediction model $\mathcal{M}$ which autoregressively predicts the next sequence element out of V possible elements y through sampling of some probability distribution with temperature τ $p(y;\tau)$. Denote the second highest probability as $p_2(y;\tau)$. Assume we only predict one next element in the sequence and have an ideal judge model J' that produces $J'(y_i, y_j) = 1$ if $i = j$ and $J'(y_i, y_j) = 0$ if $i \neq j$. If TRUST is formed through a single pairwise comparison between IID sampled next sequence elements, then*

$$E[\text{TRUST}] = (\max_i(p_i(y;\tau)))^2 + O(p_2^2(y;\tau)) \geq (\max(p(y;\tau)))^2 \tag{2}$$

Proof. Given a single next-element prediction, the probability that the two elements y_i, y_j sampled are identical is

$$p(y_i = y_j) = \sum_{i=1}^{V} p(y_i)^2 \tag{3}$$

Because our judge model is ideal, this is also the probability that the judge will return a score of 1. Thus, we have

$$E[\text{TRUST}] = p(J'(y_i, y_j) = 1) \tag{4}$$

$$= \sum_{i=1}^{V} p(y_i)^2 = (\max_i(p_i(y;\tau)))^2 + O(p_2^2(y;\tau)) \geq (\max_i(p_i(y;\tau)))^2 \tag{5}$$

Thereby proving our result. □

Evidently, TRUST scores applied only to next-token prediction are generally the square of the MSP score in addition to higher order terms, but are always lower bounded by the square MSP. In situations where the MSP is high (which is common), the TRUST score is a close approximation of the square MSP score. Even low MSP is often caused by synonyms or other semantically close tokens, and Theorem 1 would still hold with the slight modification that synonym tokens are clustered and their sampling probabilities considered jointly.

We emphasize that the theory presented here only applies to single-token prediction, and TRUST scores are even more powerful in multi-token settings (which we will demonstrate empirically later). We only provide this theory to show that TRUST scores are well-motivated by strong industry-standard uncertainty baselines.

A.3 Choosing Temperature

All of our experiments within this work were performed at a set temperature $\tau = 1.0$. We did perform some experiments at other temperatures; for example, in Figure 3 you can see the same experiments on the MATH dataset as in Figure 2 but done at different temperatures of $\tau = 0.3$ and $\tau = 0.6$. In general, the discriminative ability of TRUST seems to be fairly consistent across different temperatures. As such, we picked $\tau = 1.0$ as we knew the temperature will be high enough to induce substantial variability in the response.

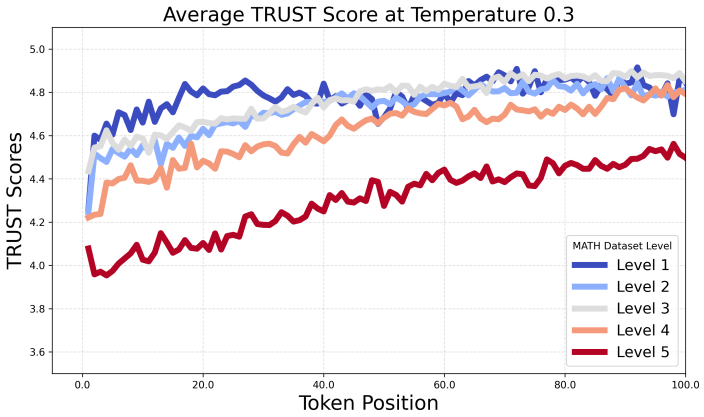
In the future, it would be interesting to also see if extensions to TRUST that take information at multiple temperatures into account can perform better. For example, it is clear from Figure 3 that higher temperatures induce lower similarity scores, which is reasonable given that higher temperatures would cause a larger degree of model divergence. Is there a signal hidden within the speed of emergence of this variability as temperature is tuned upwards that we could utilize? These types of second-order temperature effects were out of scope for the current work, where we wanted to exhibit the pure performance of single TRUST scores, but would be interesting avenues for followup research.

A.4 Simple Uncertainty Prediction

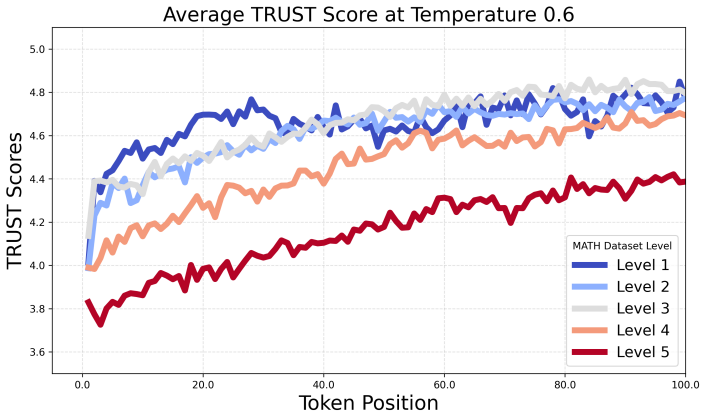
For a controlled setting, we generate a synthetic dataset of 80 simple questions and ten distinct responses to each question. These responses are preferences on a variety of topics, such as pets, outdoor activities, and cooking recipes. We will make the full dataset available at a later time, and we display an example from the dataset in Appendix A.5.

We then form ten datasets of mixing factor $M \in (0, \ldots, 9)$. A dataset with mixing ratio $M = i$ is composed of $(100 - 10M)\%$ of the first possible response to each question, with the remaining dataset split uniformly across all other responses. This means increasing M increases the dataset entropy until saturating at the uniformly distributed limit.

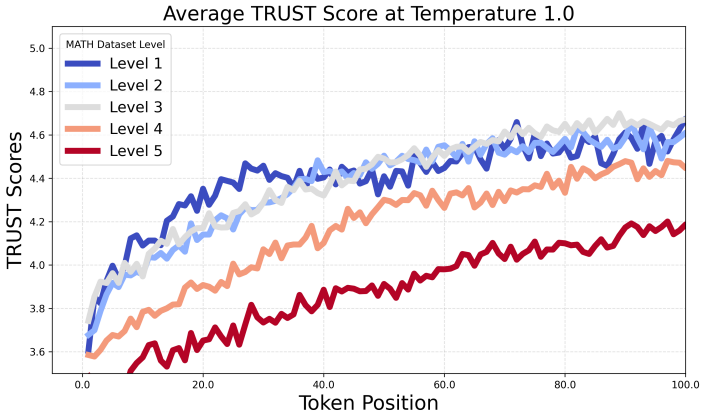
Each dataset of different mixing factor is then fine-tuned into a medium-sized LLM using LoRA (Hu et al., 2021) adapters in ten isolated trials over the ten resampled datasets respectively. Specifically, we fine-tune the Llama 3.1 8B model using LoRA adapters with the standard next-token prediction objective. We then take the trained models and test how the various baselines described in Section 4.1 correlate with the mixing factor M. Final uncertainty scores for each baseline were calculated as an average of token-level uncertainty scores across a contiguous window of token positions within the response; this window was treated as a hyperparameter and tuned for each baseline.



(a) TRUST scores for the MATH dataset at temperature 0.3.



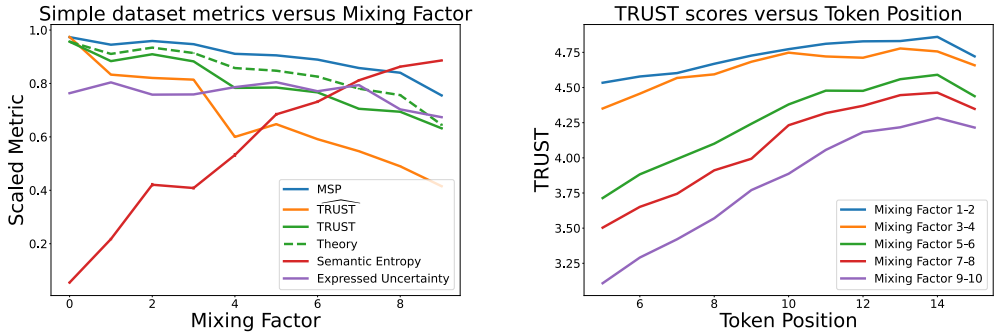
(b) TRUST scores for the MATH dataset at temperature 0.6.



(c) TRUST scores for the MATH dataset at temperature 1.0.

Figure 3: Comparing TRUST Scores across different temperature settings of LLM's.

The results are shown in Table 4. We note that the trained model $\widehat{\text{TRUST}}$ performs nearly as well as the raw TRUST scores and the Semantic Entropy scores, which shows that TRUST scores can be treated as training targets and distilled into efficient models. We also include MSP^2 as a baseline following the discussion in Appendix A.2. In this case, all methods perform fairly well in this simple problem setting, but TRUST outperforms all other baselines except for semantic entropy which performs at parity, which we expected as semantic entropy is optimal in settings with simple



(a) TRUST scores by mixing factor.

(b) TRUST scores vs token position.

Figure 4: (a) MSP and TRUST values vs mixing factor, along with a theory line at MSP^2. All metrics are scaled to lie on a [0,1] scale. (b) Average raw TRUST scores plotted against token position. We consolidate data in every other level to make the visualization smoother. As expected, TRUST scores tend to be monotonically increasing as fixing more tokens in the response leads to fewer opportunities for response branching.

Correlation $\lvert\sigma\rvert$ to Mixing Factor ↑						
$\widehat{\text{TRUST}}$	TRUST	SE	MSP	MSP^2	Entropy	EU
0.969±0.001	**0.976±0.001**	**0.977±0.001**	0.935±0.003	0.947±0.003	0.956±0.002	0.548±0.008

Table 4: Correlation coefficients of candidate uncertainty metrics vs Mixing Factor. SE is Semantic Entropy and EU is Expressed Uncertainty. MSP-Entropy is omitted with value 0.900 ± 0.006.

declarative outputs. We note that even though in certain limits we know TRUST is related to MSP^2, even in this simple setting with multi-token outputs we already outperform the single-output methods.

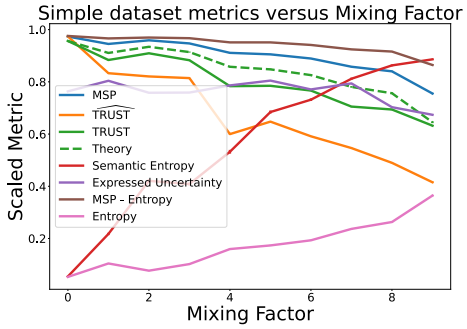
We note that correlations can depend on the functional form of the correlants: correlating x with y will give different results than correlating x^2 with y. In our case, the comparisons in Table 4 are valid to leading order because our MSP scores are generally close to 1 (See Figure 4(a)) and thus all candidate metrics admit a linear approximation. However, we do include comparisons to MSP^2 and MSP-minus-Entropy, because TRUST is quadratic in MSP (Appendix A.2) and MSP-Entropy$= MSP + \sum_i p_i \log(p_i) \approx$ MSP+MSP(MSP-1) = MSP^2. TRUST still outperforms all transformed versions of these metrics.

We display average metrics in Figure 4, along with a theory line in Figure 4(a) to show that the theory posited in Appendix A.2 tracks very closely with our observed TRUST scores. We note that even though average TRUST and the theory line are on top of each other, individual TRUST scores still outperform MSP^2 which indicates that the TRUST scores are capturing some additional semantics of the problem that single-prediction methods cannot capture.

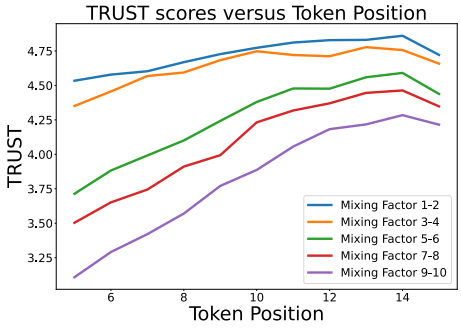
In Figure 5(a) we show a more complete version of Figure 4. The scores shown on the left are raw unscaled scores. We see that at least visibly by eye, both MSP and entropy contain more noise than TRUST , which is consistent with our observation that TRUST tends to overperform on this baseline (Table 4).

A.5 Simple human preference dataset

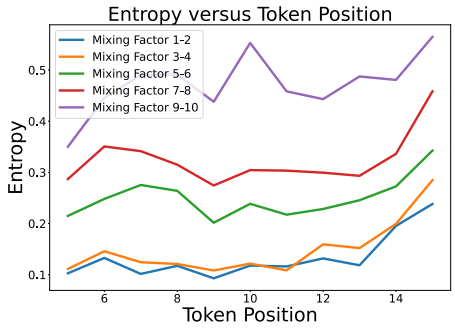
Here is an example from the simple preference dataset, which we will make available at a later time:



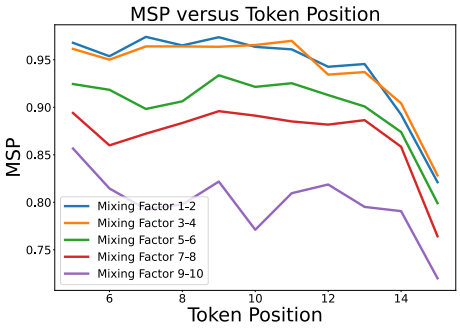
(a) All scores for each mixing factor.



(b) TRUST scores at each token position.



(c) Entropy at each token position.



(d) MSP at each token position.

Figure 5: Comparing different measures of uncertainty across mixing factors and token positions on the simple dataset.

Q: What's your favorite pet?

- I prefer parrots because they are colorful and can mimic sounds.
- I love rabbits because they are gentle and easy to care for.
- I'm fond of turtles because they are quiet and have a long lifespan.
- I enjoy having fish as pets because they are calming to watch and require minimal interaction.
- I like hamsters because they are small, friendly, and fun to watch run on their wheels.
- I favor guinea pigs because they are social creatures that enjoy interaction and have distinct squeaky voices.
- My favorite pet is a lizard because they are low-maintenance and have fascinating behaviors.
- I adore dogs because they are affectionate and enjoy outdoor activities.
- I prefer cats because they are independent and love to explore their surroundings.
- My favorite pet is a ferret because they are playful and curious, constantly exploring their environment.

The following is the system prompt used with the OpenAI gpt-4o chat completions API when generating questions sequentially to avoid repeating previously generated questions:

> Ask me a short and simple question about my preferences without giving me any options.
> The question should be less than 20 words.
> Do not try to answer the question.
> Below are questions that have been asked before. Please generate a new unrelated question.
> {previous_questions}

The following is the system prompt used with the OpenAI gpt-4o chat completions API when generating answers sequentially to avoid repeating previously generated answers:

> Given the following question and optional reference answers, create a new version of the answer with altered preferences that is not similar to the reference answers.
> The answer should be a single complete sentence with simple reasons.
> Make sure to respond with less than {word_count} words.
> Example:
> - Question: What's your favorite pet?
> - Reference answers: I like dogs because they are loyal and playful.
> - New answer: I like cats because they are independent and cuddly.
> Question:
> {question}
> Reference answers:
> {answers}

A.6 Generating TRUST completions with the completions API

The following is the instruction prepended to the question when generating completions in the **MATH dataset**:

> Continue generating the response given the following context, question and partial answer such that the total answer is at most {max_words} words.

The following is the instruction prepended to the question when generating completions in the **simple dataset**:

> Continue generating the response given the following question and partial answer such that the total answer is at most {max_words} words. The answer should be a single complete sentence with simple reasons.

A.7 Computing TRUST Scores with an LLM judge

The following is the system prompt used to judge the semantic similarity score between two completions for the **simple dataset** with the OpenAI gpt-4o model:

> **Task:** Rate the semantic similarity between Answer 1 and Answer 2 on a scale of 1-5, where:
> - 1 = Not similar meanings
> - 2 = Similar meanings with some potential dissimilarities
> - 3 = Very similar meanings with slight differences
> - 4 = Almost the same meaning with very few differences
> - 5 = Essentially the same meaning, highly semantically similar
> Answer 1:
> {answer_one}
> Answer 2:
> {answer_two}

	Similarity scores distribution				
Model	1	2	3	4	5
gpt-4-turbo	0.144	0.336	0.074	0.101	0.345
gpt-4.1	0.142	0.342	0.070	0.100	0.346
gpt-4.1-mini	0.142	0.335	0.072	0.101	0.350
gpt-4.1-nano	0.148	0.334	0.069	0.104	0.345
gpt-4o	0.142	0.345	0.069	0.097	0.347
gpt-4o-mini	0.146	0.334	0.075	0.098	0.347
gpt-5	0.143	0.337	0.075	0.099	0.346
gpt-5-mini	0.138	0.345	0.068	0.104	0.345
gpt-5-nano	0.147	0.329	0.081	0.094	0.349
llama-3.1-70b	0.143	0.336	0.074	0.098	0.349
llama-3.1-8b	0.148	0.333	0.078	0.093	0.348
llama-3.3-70b	0.144	0.335	0.077	0.093	0.351

Table 5: Similarity scores distribution for different models.

	Similarity scores distribution				
Trials	1	2	3	4	5
5	0.132	0.338	0.064	0.100	0.366
10	0.142	0.345	0.069	0.097	0.347
15	0.137	0.365	0.061	0.097	0.341
20	0.138	0.358	0.061	0.098	0.346
25	0.132	0.355	0.061	0.095	0.357
30	0.129	0.357	0.061	0.096	0.357
35	0.133	0.352	0.063	0.097	0.355
40	0.133	0.350	0.063	0.097	0.357
45	0.132	0.348	0.067	0.096	0.357
50	0.129	0.348	0.067	0.097	0.359

Table 6: Similarity scores distribution for different number of trials.

The following is the system prompt used to judge the semantic similarity score between two completions for the **MATH dataset** with the OpenAI gpt-4o model:

```
**Task:** Rate the semantic similarity between Answer 1 and Answer 2 on a scale of 1-5,
where:
- 1 = Completely different meanings, no semantic overlap
- 2 = Mostly different with minimal semantic similarity
- 3 = Some semantic similarity, but notable differences in meaning
- 4 = Very similar meanings with minor differences
- 5 = Essentially the same meaning, highly semantically similar
Answer 1:
{answer_one}
Answer 2:
{answer_two}
```

In Table 5 we show how changing the LLM judge model while keeping the number of trials N = 10 and the system prompt fixed doesn't influence the distribution of 1-5 similarity scores significantly over the last 100 examples of the MATH dataset.

In Table 6 we show how increasing the number of trial rollouts keeping the gpt-4o judge model and the system prompt fixed doesn't influence the distribution of 1-5 similarity scores significantly above the current value of N = 10 trials over the last 100 examples of the MATH dataset.

A.8 REGENERATING Q&A RESPONSES AND HALLUCINATION LABELS

For our hallucination dataset from Section 4.4 we regenerated responses as well as hallucination labels at the token-level.

Here is the system prompt we used for generating responses using gpt-4o-mini through the completions API for a given question in consideration of any pre-existing knowledge chunks from the original datasets.

```
Knowledge:
{knowledge}
Question:
{question}
Answer:
```

Here is the system prompt we used for generating hallucination labels using gpt-4o for each completion at each token position:

```
You are a highly knowledgeable and careful evaluator of question-answering, dialogue and
summarization systems. Given some optional knowledge, a question and a response, your
task is to judge whether the response contains any hallucination — that is, information
unsupported, contradicted, or absent from the provided knowledge, or factually incorrect in
general.
Assess the extent of hallucination based only on the given knowledge, not on your own
external knowledge.
Assign a hallucination severity integer score between 1 and 5 only for the response, using the
following criteria:
**1 - No Hallucination (Fully Grounded)** The response is entirely accurate and supported
by the provided knowledge. All claims, facts, and details are either directly stated in or
logically derivable from the knowledge base. The response adds no unsupported information
and maintains complete fidelity to the source material.
**2 - Minimal Hallucination (Minor Embellishment)** The response is largely accurate with
only trivial deviations. It may include minor stylistic flourishes, reasonable inferences that
don't alter meaning, or negligible details that are commonly accepted as true. These additions
do not meaningfully impact the core accuracy or trustworthiness of the response.
**3 - Moderate Hallucination (Noticeable Fabrication)** The response contains noticeable
unsupported information that goes beyond the provided knowledge. This may include
plausible-sounding details that aren't verified, moderate speculation presented as fact, or
statements that conflate or misrepresent some aspects of the knowledge. While parts of the
response remain accurate, the fabricated elements are significant enough to reduce reliability.
**4 - Severe Hallucination (Major Inaccuracies)** The response contains substantial fabri-
cated or incorrect information that significantly undermines its trustworthiness. Key facts
may be misrepresented, contradicted, or invented outright. While some correct information
may remain, the hallucinated content is pervasive enough that the response could mislead
users on important points.
**5 - Complete Hallucination (Wholly Fabricated)** The response is predominantly or
entirely fabricated, with little to no grounding in the provided knowledge. It may directly
contradict the source material, invent major facts out of whole cloth, or be fundamentally
disconnected from the question and knowledge provided. The response is unreliable and
potentially harmful in its inaccuracy.
Knowledge:
{knowledge}
Question:
{question}
Answer:
{answer}
```

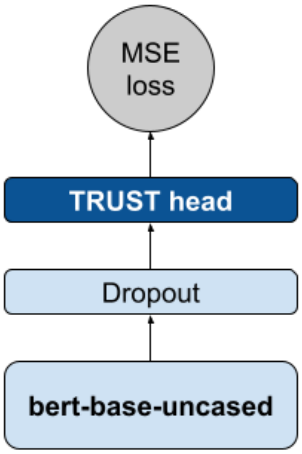


Figure 6: BERT model architecture with TRUST prediction head

We obtained this system prompt with gpt-5.1 through a manual iteration approach. We provide the previous version of the system prompt together with measured frequencies of each level of hallucination throughout a subset of generated completions as input to gpt-5.1. We ask the LLM to output a new version where the over-represented and under-represented levels are better calibrated, effectively softening the criteria where levels are over-represented and hardening the criteria where levels are under-represented.

We plan to make the dataset available at a later time. Here is an example from the dataset:

> Knowledge:
> Storm Front is written by Jim Butcher. Jim Butcher wrote Turn Coat
> Question:
> [Human]: Could you recommend a book by the author of Storm Front?
> [Assistant]: Sure! That would be written by Jim Butcher, who also wrote Turn Coat. Have you read that one?
> [Human]: No I have not. What genre is it?
> [Assistant]: It's an urban fantasy as well as mystery and speculative fiction.
> [Human]: Sounds intriguing. Do you know the year it was published?
> Answer:
> [Assistant]: Yes, Turn Coat was published in 2009.

A.9 TRUST BERT MODEL ARCHITECTURE

The high level architecture of the $\widehat{\text{TRUST}}$ distilled model is in Figure 6, which shows a standard BERT trunk coupled with our additional TRUST logistic head which predicts TRUST scores.

Here follows the detailed description of the BERT model including the additional logistic head which predicts TRUST scores:

```
BertTokenSimilarityModel(
  (bert): BertModel(
    (embeddings): BertEmbeddings(
      (word_embeddings): Embedding(30522, 768, padding_idx=0)
      (position_embeddings): Embedding(512, 768)
      (token_type_embeddings): Embedding(2, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (encoder): BertEncoder(
```

```
      (layer): ModuleList(
        (0-11): 12 x BertLayer(
          (attention): BertAttention(
            (self): BertSdpaSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
            (output): BertSelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
          (intermediate): BertIntermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): BertOutput(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
      )
    )
    (pooler): BertPooler(
      (dense): Linear(in_features=768, out_features=768, bias=True)
      (activation): Tanh()
    )
  )
  (dropout): Dropout(p=0.1, inplace=False)
  (trust_head): Linear(in_features=768, out_features=128, bias=True)
)
```

A.10 BASELINE METHODS

Here we provide some additional implementation details for all our baseline methods. In terms of calculating some of our white-box statistics like MSP and entropy, conventional LLM APIs can only return a truncated list of log-probabilities containing up to k out of the supported vocabulary V. This list is limited to the top 5 log-probabilities from the Fireworks AI API, which we used to query Llama models, or the top 100 log-probabilities from the OpenAI API.

Let $k \in \{5, 100\}$ be the number of log-probabilities returned by these APIs respectively and y_i be the i-th probability in the returned list of top probabilities for any generated token.

After converting the returned log-probabilities to the original softmax values through exponentiation, one can compute the following Partial Entropy for any generated token:

$$PE_k = -\sum_{i=1}^{k} p(y_i;\tau) \cdot log\Big(p(y_i;\tau)\Big) \tag{6}$$

LLMs attribute most of the probability mass to these top k tokens, as evidenced by the high MSP scores we observed throughout all our experiments, so we assume for simplicity with no alternative that the remaining entries that were not returned by LLM APIs follow a uniform distribution. We denote the Residual Probability by:

$$RP_k = 1 - \sum_{i=1}^{k} p(y_i; \tau) \tag{7}$$

Then we divide the residual probability mass among the remaining $(V - k)$ tokens equally to obtain the Residual Entropy:

$$RE_k = -\sum_{i=1}^{V-k} \frac{RP_k}{V-k} \cdot log\left(\frac{RP_k}{V-k}\right) = -RP_k \cdot \ log\left(\frac{RP_k}{V-k}\right) \tag{8}$$

Lastly, combine the Partial Entropy with the Residual Entropy to obtain the Estimated Entropy:

$$EE_k = PE_k + RE_k \tag{9}$$

Trivially, the Maximum Softmax Probability is represented by:

$$MSP = \max_i p(y_i; \tau) \tag{10}$$

Finally, the Ensemble KL Divergence is the log-adjusted pairwise mean of KL divergences among h number of ensemble prediction head distributions with head indices $a \neq b$:

$$EKL = \binom{h}{2}^{-1} log\left(\sum_{a \neq b}^{\binom{h}{2}} D_{KL}\Big(p(y^a; \tau) \mid\mid p(y^b; \tau)\Big)\right) \tag{11}$$