# Uncertainty for Language Model

Hangyue Li

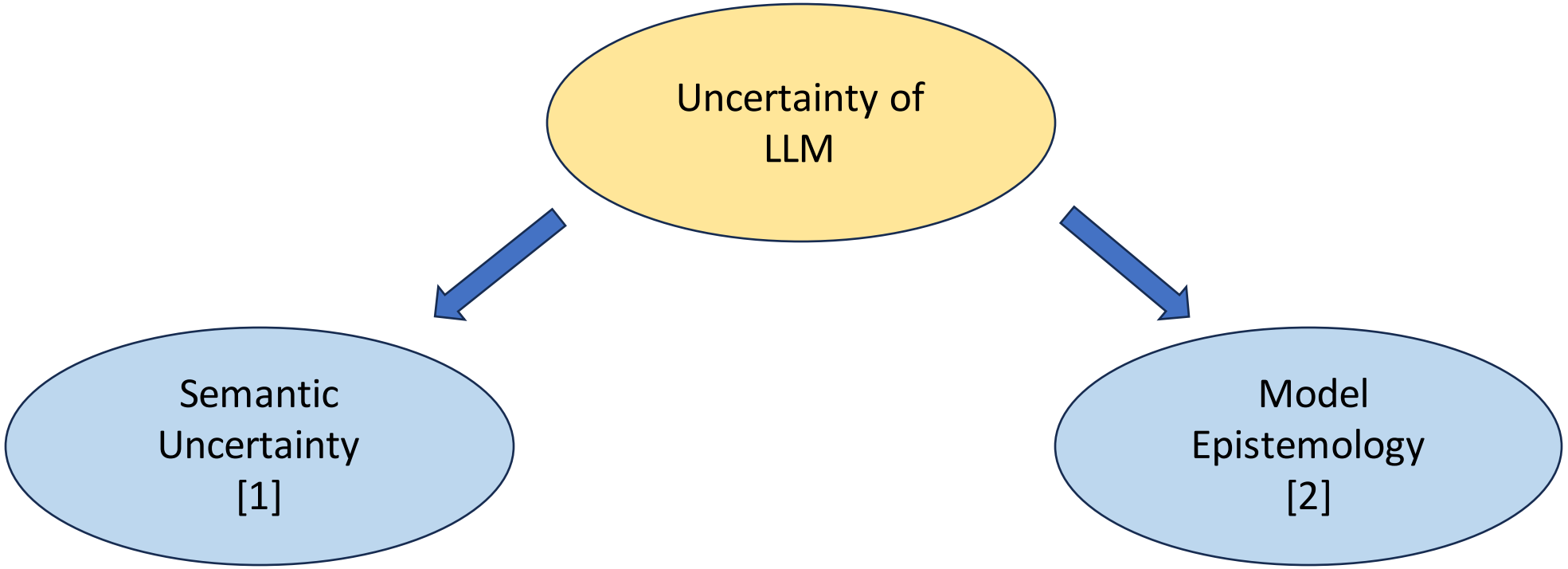
Feb 13 2024

# Uncertainty in large language models

- The measure of how confident a model is about the predictions it makes.

- Providing insight into how much trust can be placed in outputs.
  - 1. Enhancing Model Reliability
  - 2. Safety and Risk Management
  - 3. Model improvement

# Uncertainty in large language models




- [1] Kuhn, Lorenz, Yarin Gal, and Sebastian Farquhar. "Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation." *arXiv preprint arXiv:2302.09664* (2023)

- [2] Navigating the Grey Area: How Expressions of Uncertainty and Overconfidence Affect Language Models https://aclanthology.org/2023.emnlp-main.335 Zhou et al., EMNLP 2023

# Uncertainty: Background

- Probabilistic tools for uncertainty estimation:

- The total uncertainty of a prediction can be understood as the predictive entropy of the output distribution.

- The predictive entropy for a point x is the conditional entropy of the output random variable Y with realisation y given x

$$PE(x) = H(Y \mid x) = -\int p(y \mid x) \ln p(y \mid x) dy \qquad (1)$$

# Semantic Uncertainty

- Motivation: when we can trust the natural language outputs of models

- Challenge: measuring uncertainty in natural language is challenging because of 'semantic equivalence'
  - different sentences can mean the same thing.
  - "France's capital is Paris" or "Paris is France's capital"

Table 3: Illustration of semantic, syntactic, and lexical equivalence. Work with foundation models implicitly focuses on *lexical* equivalence, which entails the others, but we usually care about *semantic* equivalence.

| | | Equivalence | | |
|---|---|---|---|---|
| Sentence A | Sentence B | Lexical | Syntactic | Semantic |
| Paris is the capital of France. | Paris is the capital of France. | ✓ | ✓ | ✓ |
| | Berlin is the capital of France. | | ✓ | |
| | France's capital is Paris. | | | ✓ |

# Semantic Equivalence

- Require that the sequences mean the same thing with respect to the context
- Calculate the entropy based on semantic equivalence

Table 1: Answers to the question "What is the capital of France?" (a) When all generations from the model mean different things, semantic clustering has no effect—the entropy and semantic entropy are identical. (b) When some of the answers are semantically equivalent ("Paris" and "It's Paris") the semantic entropy does a better job of capturing the actually low uncertainty.

(a) Scenario 1: No semantic equivalence

| Answer $\mathbf{s}$ | Likelihood $p(\mathbf{s} \mid x)$ | Semantic likelihood $\sum_{\mathbf{s} \in c} p(\mathbf{s} \mid x)$ |
|---|---|---|
| Paris | 0.5 | 0.5 |
| Rome | 0.4 | 0.4 |
| London | 0.1 | 0.1 |
| Entropy | 0.94 | 0.94 |

(b) Scenario 2: Some semantic equivalence

| Answer $\mathbf{s}$ | Likelihood $p(\mathbf{s} \mid x)$ | Semantic likelihood $\sum_{\mathbf{s} \in c} p(\mathbf{s} \mid x)$ |
|---|---|---|
| **Paris** | 0.5 | 0.9 |
| **It's Paris** | 0.4 | |
| London | 0.1 | 0.1 |
| Entropy | 0.94 | 0.33 |

# Semantic Equivalence: Key Challenge

- Recall the outputs of models:
  - token-likelihoods -- representing lexical confidence.

$$p(\mathbf{s} \mid x) = \prod_i p(s_i \mid s_{<i}, x)$$

P(Sally, fed, my, cat, with, meat) = P(Sally)
* P(fed | Sally)
* P(my | Sally, fed)
* P(cat | Sally, fed, my)
* P(with | Sally, fed, my, cat)
* P(meat | Sally, fed, my, cat, with)

  - But we care about meanings!
  - Yet, at a token-level the model could be uncertain between two forms of the same meaning
  - “France’s capital is Paris” or “Paris is France’s capital” is not uncertain

# Semantic Equivalence: How?

- A placeholder semantic equivalence relation: $E(\cdot,\cdot)$,
  - That is, for the space of semantic equivalence classes $C$, the sentences in the set $c \in C$ all share a meaning such that $\forall s, s_0 \in c : E(s, s_0)$.

- Probability of the model generating any sequence that shares same meaning:

$$p(c \mid x) = \sum_{\mathbf{s} \in c} p(\mathbf{s} \mid x) = \sum_{\mathbf{s} \in c} \prod_i p(s_i \mid s_{<i}, x). \tag{2}$$

- Semantic entropy (SE) as the entropy over the meaning-distribution

$$SE(x) = -\sum_c p(c \mid x) \log p(c \mid x) = -\sum_c \left( \left( \sum_{\mathbf{s} \in c} p(\mathbf{s} \mid x) \right) \log \left[ \sum_{\mathbf{s} \in c} p(\mathbf{s} \mid x) \right] \right). \tag{3}$$

# Semantic Equivalence: How?

- Examines uncertainty in meaning-space

- At a high level this involves three steps:

1. **Generation:** Sample $M$ sequences $\{s^{(1)}, \ldots, s^{(M)}\}$ from the predictive distribution of a large language model given a context $x$.
2. **Clustering:** Cluster the sequences which mean the same thing using our bi-directional entailment algorithm.
3. **Entropy estimation:** Approximate semantic entropy by summing probabilities that share a meaning following Eq. (2) and compute resulting entropy.

# Clustering by semantic equivalence

- DeBERTa-large model [1] that is fine-tuned on the NLI data set MNLI [2]
  - concatenate each of the two question/answer pairs. The Deberta model then classifies this sequence into one of: entailment, neutral, contradiction.
  - compute both directions $(s, s_0)$ and $(s_0, s)$, and the algorithm returns equivalent if both directions were entailment

[1] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. arXiv preprint arXiv:2006.03654, 2020a. 5
[2] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. arXiv preprint arXiv:1704.05426, 2017. 5

# Experiment setup

- Model: GPT-like OPT models [1] varying the size of the model between 2.7B, 6.7B, 13B and 30B parameters
  - No ensembling and no stochastic or Bayesian modification
- Datasets:
  - CoQA Reddy as an open-book conversational question answering problem (the model answers a question using a supporting paragraph)
  - TriviaQA as a closed-book QA problem (the model must answer a question without access to a supporting paragraph)
- Baselines:
  - Predictive entropy
  - Length-normalised predictive entropy
  - p(True): estimate by ‘asking’ the model if its answer is correct
  - Lexical similarity

# Semantic Equivalence: Empirical Evaluation

- Effective uncertainty measures should offer information about how reliable the model's answers are
  - very uncertain generations should be less likely to be correct

- Evaluate uncertainty as the problem of predicting whether to rely on a model generation for a given context
  - whether to trust an answer to a question. (Binary classification: if the answer is correct)

  - Metric: AUROC

# Semantic Equivalence: Empirical Evaluation

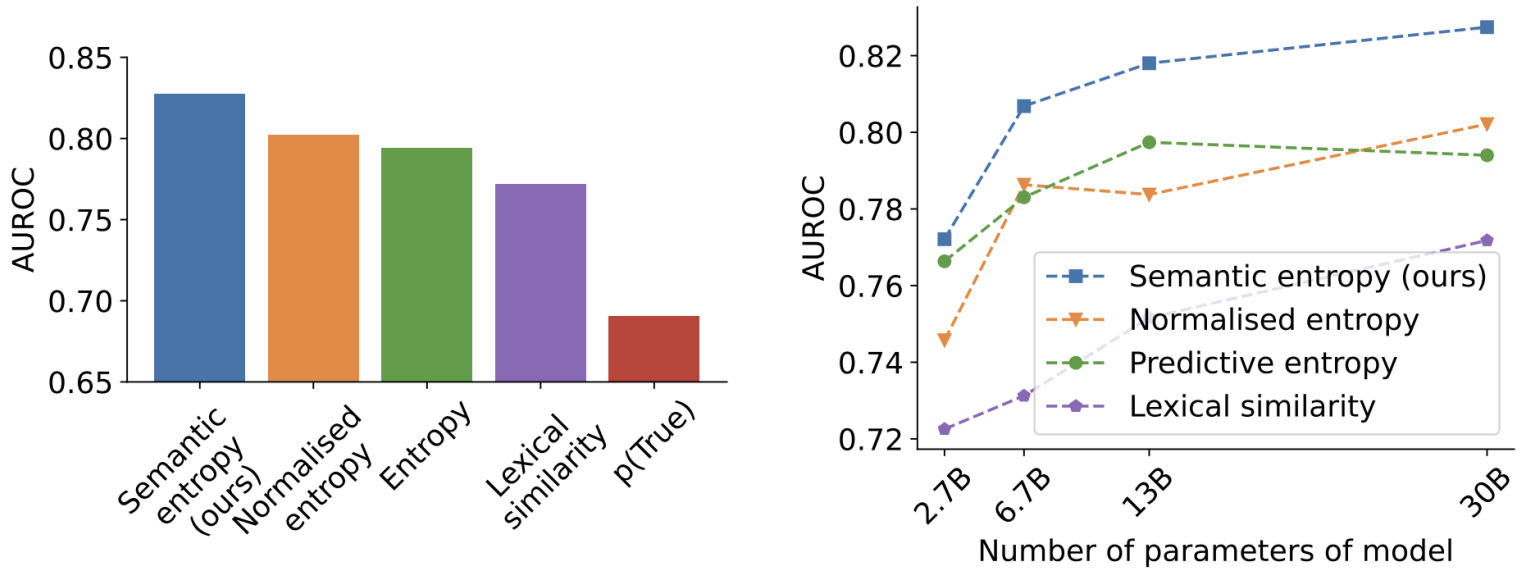



Figure 1: (a) Our semantic entropy (blue) predicts model accuracy better than baselines on the free-form question answering data set TriviaQA (30B parameter OPT model). Normalised entropy reimplements single-model variant of Malinin & Gales (2020), lexical similarity measures the average Rouge-L in a sampled set of answers for a given question analogously to Fomicheva et al. (2020), entropy and $p(\text{True})$ reimplement Kadavath et al. (2022). (b) Our method's outperformance increases with model size while also doing well for smaller models.

# Semantic Equivalence: Empirical Evaluation

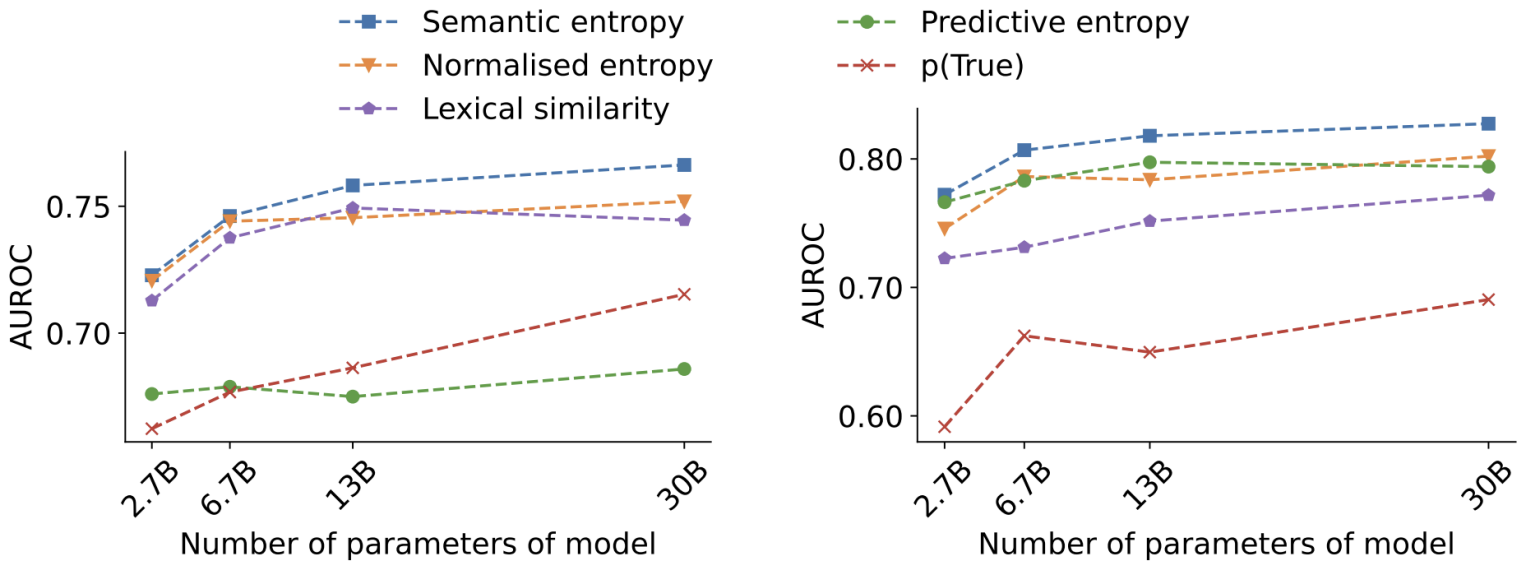



Figure 2: (a) On CoQA open-book question answering semantic entropy demonstrates better uncertainty than ordinary predictive entropy with and without normalisation at larger model sizes. It also performs significantly better than $p(\text{True})$. (b) TriviaQA shows similar results. Identical to Fig. 1b with the addition of $p(\text{True})$, which was previously omitted to avoid stretching the scale.

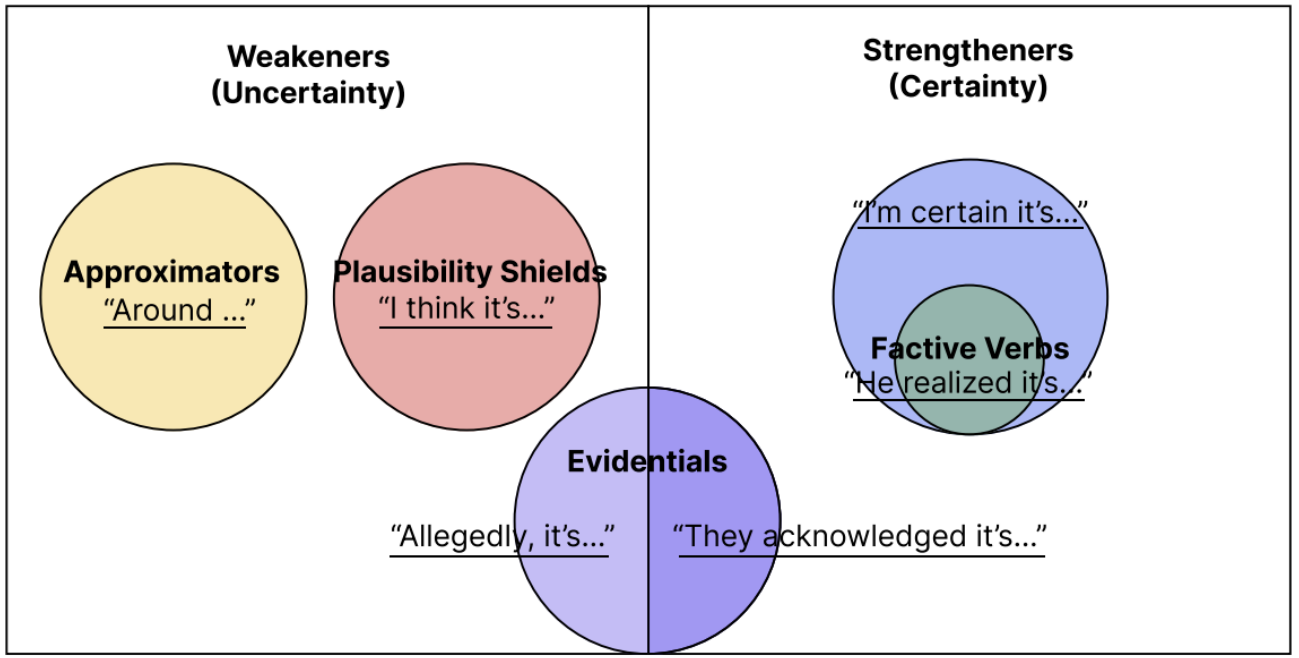
# Discussions

- Strength
  - The high-level idea of semantic entropy in the meaning space is quite reasonable for natural language generation
- Weakness
  - The semantic entropy is based on the model of DeBERTa, heavily depending on the performance of such models.
  - Probably more direct approaches to evaluate uncertainty metrics.

# Navigating the Grey Area: How Expressions of Uncertainty and Overconfidence Affect Language Models

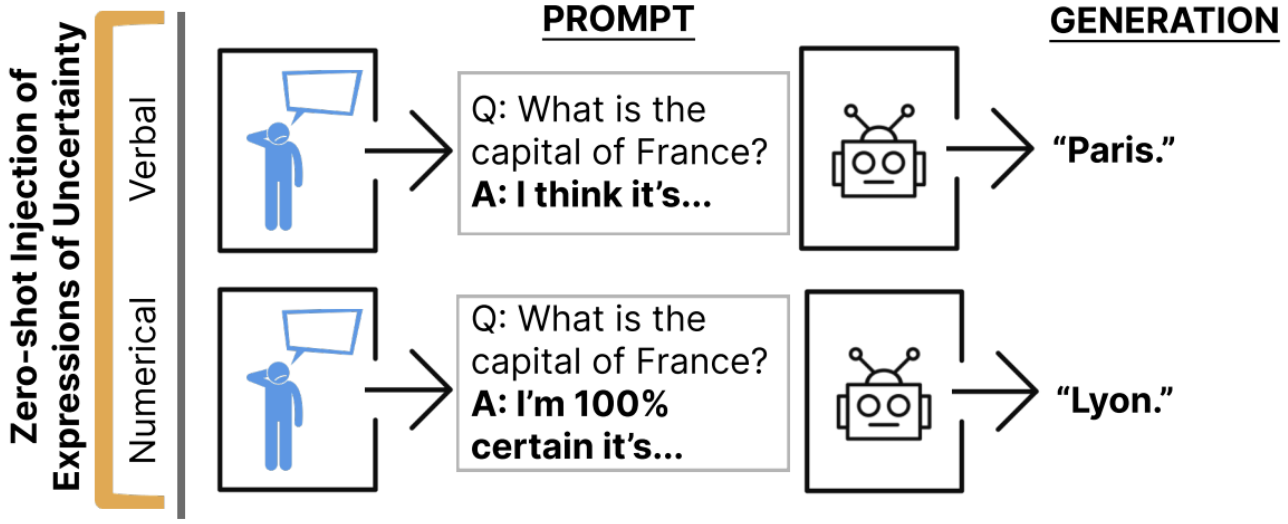
- Motivation: how epistemic markers affect the model performance?

- Epistemic markers: expressions of uncertainty



# Expressions of Uncertainty: Methods

- Inject markers into prompts for question answering
  - using zero-shot promoting to inject verbal and numerical uncertainties into trivia questions
  - measuring how language generation varies when prompted with expressions of uncertainty



# Experiment settings

- Datasets:
  - TriviaQA
  - Natural Questions (closed-book)
  - CountryQA

- Create fifty sentences (minimal pairs) for every question

- For the generated tokens, take the sum of the probability assigned to the gold answer(s) to be the probability-on-gold.
- Calculating accuracy, generate 10 tokens and if any of the tokens match the answer.

# Impact of Uncertainty

- The first hypothesis: models are robust to added expressions of uncertainty in the prompt.

- Second hypothesis: a marker suggesting certainty might be more likely to produce the correct response than a prompt with low certainty.

# Impact of Uncertainty

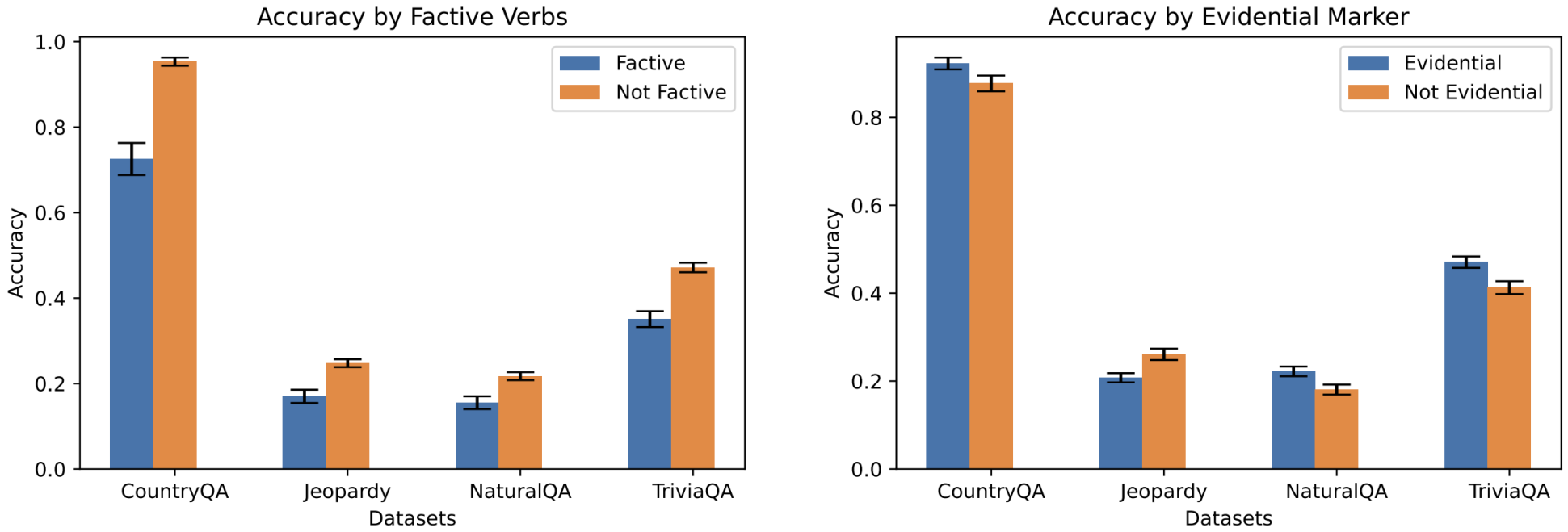



Figure 3: Significant and consistent accuracy losses for templates with factive verbs (left). Evidential markers significantly improves accuracy in three out of four datasets (right). 95% CI calculated using bootstrap resampling. Visualizing results for GPT-3 (davinci).

Factive verbs: presuppose certainty or truth, e.g. "know", "realize", or "understand", "X realizes Y"

Evidential markers: tells where the information came from, e.g., "According to research in the latest issue of Nature", "Two recent studies demonstrate that. . . "

# Impact of Uncertainty

- Certainty actually hurts accuracy!

| | ada | babbage | curie | davinci | instruct | gpt-4 |
|---|---|---|---|---|---|---|
| Boosters | 0.091 | 0.257 | 0.313 | 0.392 | 0.589 | 0.793 |
| Hedges | 0.079 | 0.272 | **0.333*** ** | **0.468*** ** | **0.642*** ** | **0.822*** ** |
| Factive Verbs | 0.078 | 0.237 | 0.293 | 0.347 | 0.555 | 0.771 |
| Non-Factives Verbs | **0.085*** | **0.276*** ** | **0.336*** ** | **0.468*** ** | **0.641*** ** | **0.821*** ** |
| Evidentials | **0.087** ** | **0.281*** ** | **0.347*** ** | **0.449*** | **0.640*** ** | **0.820*** ** |
| Non-evidentials | 0.080 | 0.250 | 0.301 | 0.433 | 0.601 | 0.799 |

Table 1: Across all six models tested, hedges outperform boosters, non-factive verbs outperform factives and evidentials out-perform non-evidentials. (Instruct = `text-davinci-003`, `GPT4` uses context window 32K.) *t*-test *p*-values, * < 0.05, ** < 0.01, *** < 0.001**.

| **Expressions** | **uncertainty** |
|---|---|
| i think | hedge |
| it could be | hedge |
| it might be | hedge |
| maybe it's | hedge |
| it should be | hedge |
| i know | booster |
| i'm certain | booster |
| i am certain | booster |
| i'm sure | booster |
| i am sure | booster |
| it must be | booster |
| evidently it's | booster |

# Why Certainty Hurt?

- Certainty affects performance independent of perplexity
  - Phrases with high perplexity result in a significant drop in performance when used as prompts in language modeling.

- Weakeners led to a flattening of the distribution of probability
  - increase in accuracy of weakeners is not due to an increase in answer confidence but diversity

- Certainty used in questions instead of answers
  - language models might be mimicking this behavior and responding to prompts with epistemic markers

# Numerical Values for Uncertainty

- "I'm 90% certain. . . "
- "70% chance it's. . . "

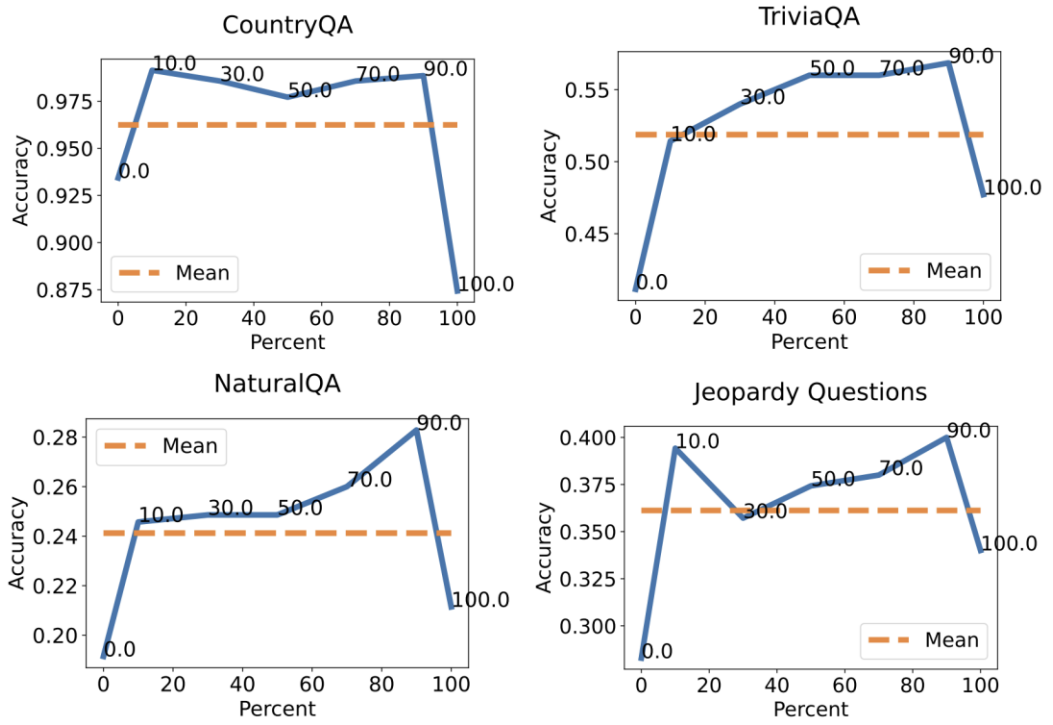



Figure 4: The X-axis indicates the percentage that was injected into the verbal uncertainty. The Y-axis indicates the accuracy across numerical uncertainties. Note the consistent drop in accuracy between 90% and 100% uncertainty and the increase in accuracy between 0% and 10% uncertainty.

# To be discussed

- Can Models Generate Expressions of Uncertainty?
  - challenging to calibrate models to generate epistemic markers.
- How to integrate attributions of information in a verified manner?
  - phrases like "Wikipedia says. . . ", however these could be falsely injected attributions.
- Syntatic, idomatic, and pragamtic differences in hedges could be interesting to study in follow-up work
  - Humans use language that contains expressions of certainty when they are, in fact, not certain, and models appear to be mimicking this behavior.