Navigating the Grey Area: Expressions of Overconfidence and Uncertainty in Language Models

Kaitlyn Zhou
Stanford University
katezhou@stanford.edu

Dan Jurafsky
Stanford University
jurafsky@stanford.edu

Tatsunori Hashimoto
Stanford University
thashim@stanford.edu

Abstract

Despite increasingly fluent, relevant, and coherent language generation, major gaps remain between how humans and machines use language. We argue that a key dimension that is missing from our understanding of language models (LMs) is the model's ability to interpret and generate *expressions of uncertainty*. Whether it be the weatherperson announcing a chance of rain or a doctor giving a diagnosis, information is often not black-and-white and expressions of uncertainty provide nuance to support human-decision making. The increasing deployment of LMs in the wild motivates us to investigate whether LMs are capable of interpreting expressions of uncertainty and how LMs' behaviors change when learning to emit their own expressions of uncertainty. When injecting expressions of uncertainty into prompts (e.g., "I think the answer is..."), we discover that GPT3's generations vary upwards of 80% in accuracy based on the expression used. We analyze the linguistic characteristics of these expressions and find a drop in accuracy when naturalistic expressions of certainty are present. We find similar effects when teaching models to emit their own expressions of uncertainty, where model calibration suffers when teaching models to emit certainty rather than *un*certainty. Together, these results highlight the challenges of building LMs that interpret and generate trustworthy expressions of uncertainty.

1 Introduction

As natural language systems are increasingly used in real-life scenarios, it is becoming important to not only produce fluent and correct answers but also to properly communicate uncertainties. From weather reports to doctors' offices, humans are constantly integrating expressions of uncertainty into decision-making processes. Expressions of uncertainty inform us of the confidence, source, and limitations of information, helping us make big and small decisions like bringing an umbrella or starting a course of chemotherapy. Despite substantial literature on methods for quantifying statistical uncertainty, there has been comparatively less attention on how such linguistic uncertainties might interact with the natural language generation system, resulting in a lack of understanding of this critical component of how models interact with natural language. Our work makes progress in answering this question by asking: are language models (LMs) capable of interpreting expressions of uncertainty and how do LMs' behaviors change when trained to emit their own expressions of uncertainty?

Naturalistic expressions of uncertainty cover a broad range of discourse acts such as signaling hesitancy, attributing information, or acknowledging limitations. While prior work has explored linguistically calibrating model generations, this has primarily focused on learning the mapping between the internal probabilities of a model and a verbal or numerical ordinal output (Kadavath et al., 2022; Lin et al., 2022; Mielke et al., 2022). Our work, by contrast, seeks to understand and incorporate non-uni-dimensional linguistic features such as hedges or epistemic markers (e.g., It could be ...), factive verbs (e.g., We realize it's...), and evidential markers (e.g., Wikipedia says it's...) — building our understanding of how these additional properties and evidentiality impact natural language generation.

We begin with an investigation of how LMs interpret uncertainty in prompts, followed by a small study on how LMs behave when generating their own uncertainties via in-context learning. We focus on studying this question in the question-answering (QA) setting, due to its real-world relevance and existing benchmarks. The first experiments are conducted in a zero-shot setting, making it possible to analyze and isolate the effects of uncertainty in prompting, while the second experiments examine how learning to express uncertainty impacts generation in QA tasks.

In both sets of experiments, we find shortcomings when expressions of high certainty are used, both in accuracy and calibration. In zero-shot prompting, we find that there are systematic losses in accuracy when expressions of certainty are used to strengthen prepositions (i.e., "We're 100% certain..."). In the second in-context learning scenario, we find that teaching the model to emit weakeners rather than strengtheners results in better calibration without sacrificing accuracy. We then discuss designing linguistically calibrated models, especially given the potential downfalls of models emitting highly certain language.

Our work offers four key contributions:

- We provide a framework and carry out analysis on how expressions of uncertainty interact with large language models.
- We introduce a typology of expressions of uncertainty to evaluate how linguistic features impact LM generation.
- We demonstrate how model accuracy suffers when models use expressions of certainty (e.g., factive verbs) or idiomatic language (e.g., 'I'm 100% certain')
- Our results in in-context learning suggest that GPT3 struggles to emit expressions of certainty in a calibrated manner but that expressions of **un**certainty might lead to better calibration.

Together, our findings illustrate the need for model analysis through the lens of uncertainty, especially as LMs become deployed in real-life decision-making settings.

2 Expressions of Certainty and Uncertainty: Linguistic Background

There is a broad literature on linguistic markers that relate to certainty and uncertainty, both for weakening category membership and weakening speaker commitment to the truth value of a proposition (see Related Work). For convenience in this paper we broadly group these linguistic devices into **weakeners** and **strengtheners**.

The most widely studied weakeners are hedges, first defined by Lakoff (1975) as related to modifying or weakening the category membership of a predicate or nominal. The most central kind of hedges are **approximators** (e.g., *somewhat*, *kind of*, *about*, *approximately*), which hedge propositional content. Another class of weakeners that some (like Prince et al. (1982)) but not others classify under hedges are **plausibility shields** (Prince et al., 1982) which express a speaker's lower level of commitment (e.g., *I think*, *I believe*).

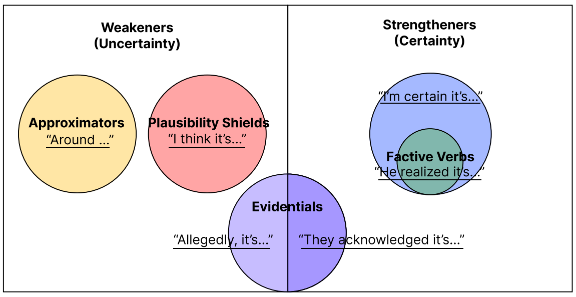


Figure 1: Lexical Features of Expressions of Uncertainty. Uncertainty classification partly adapted from (Prince et al., 1982). Certainty markers are strengtheners which contain factive verbs. Evidential markers can be both expressions of certainty and uncertainty (strengtheners or weakeners). Personal pronouns and reference to sources are additional dimensions of expressions of (un)certainty not shown in this diagram.

Strengtheners are liguistic constructions that mark *certainty*. We use the term strengtheners both to refer to strengthening speaker commitment to truth value, and to strengthening category membership. Strengtheners include **boosters** or **intensifiers** like *"I am certain"* or *"Undoubtedly"* (Hyland, 2005, 2014).

Whereas boosters can assert certainty or truth, a second kind of strengthening construction, the **factive verb**, is used to **presuppose** certainty or truth. Factive verbs like "know", "realize", or "understand" (Kiparsky and Kiparsky, 1970) presuppose the truth of the complement sentence. The statement "X realizes Y" presupposes that Y is true, (and also asserts that X is aware of it). By contrast, "X believes Y" makes no presupposition about the truth of Y. Thus the sentence "He realizes [Madrid is the capital of France]" is infelicitous, because "realize" as a factive verb presupposes the truth of its complement clause, which in this case is false (Madrid is not the capital of France).

We note the existence of a third class of markers that can be used to mark both certainty and uncertainty by directly indicating the source of the information. Such expressions (like *According to Wikipedia*, or *I heard* or *I saw*) are called **evidential marker**, linguistic signals that tells the hearer where the information came from (Aikhenvald, 2004). One subtype of evidential markers, quotative markers, are used when reported information overtly references a source (e.g., *According to research in the latest issue of Nature*, *Two recent studies demonstrate that...*). Specifically, we examine when references are citing a source (e.g., *Wikipedia says*) versus when the source is unspecified or very indirect (e.g., *They said*). We'll refer to this former case as **sourced** as a shorthand for indicating that a source is mentioned.

Finally, first-person **personal pronouns** (e.g., *I*, *we*, *our*) can be used to mark subjectivity and uncertainty in expressions like "*I think*".

2.1 Expressions of Uncertainty Typology

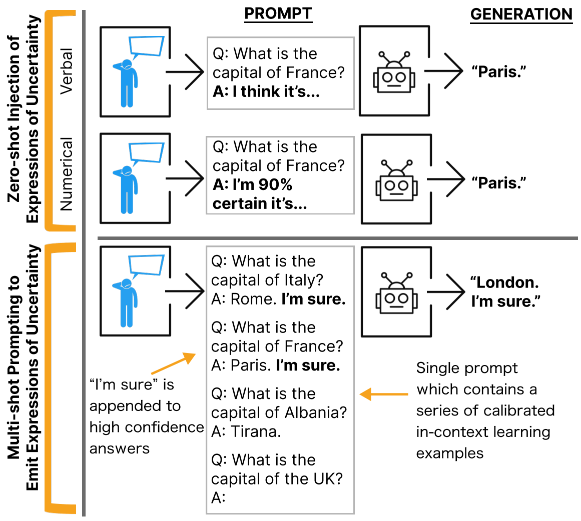
We compiled a list of fifty expressions of verbal uncertainty from both crowd-workers and the authors, focusing on the use of weakeners, strengtheners, plausibility shields, factives, evidential markers, mentions of sources, and personal pronouns. We then coded each expression with the linguistic features above. The final list (Table B) allows us to systematically analyze how expressions of uncertainty impact language modeling in the QA setting. Figure 1 shows a diagram of how each of these features relates to certainty and uncertainty in our coding scheme (details in Appendix B).

3 Methods

Our work studies LM expressions of uncertainty in the context of open-ended question answering. In both of our settings, we insert linguistic expressions of uncertainty into a QA question through handcrafted templates (Figure 2), and query the LM using this modified question answering prompt.

In the first setting, we use zero-shot promoting with expressions of uncertainty to elicit answers. We inject verbal uncertainties into trivia questions, converting a prompt like "What is the capital of France?" to, "What is the capital of France, I think it's...".

In the second setting, we create an in-context learning prompt with nearly fifty samples that contain question-answer pairs, along with an expression of (un)certainty for any (un)certain examples. For example, if the model has high confidence in an answer, the sample might appear in the in-context learning prompt as, "What's the capital of France? Paris. I'm sure". And vice versa for low-confidence answers.



Zero-shot Injection of Expressions of Uncertainty
Verbal
Numerical
Multi-shot Prompting to Emit Expressions of Uncertainty
PROMPT
GENERATION
Q: What is the capital of France? A: I think it's...
"Paris."
Q: What is the capital of France? A: I'm 90% certain it's...
"Paris."
Q: What is the capital of Italy? A: Rome. I'm sure.
Q: What is the capital of France? A: Paris. I'm sure.
Q: What is the capital of Albania? A: Tirana.
Q: What is the capital of the UK? A:
"London. I'm sure."
"I'm sure" is appended to high confidence answers
Single prompt which contains a series of calibrated in-context learning examples

Figure 2: Two key settings to analyze expressions of uncertainty. The first uses zero-shot promoting and injects verbal and numerical uncertainties into trivia questions. The second uses multi-shot in-context learning to teach models to emit expressions of certainty or uncertainty (not shown here) in its generation.

3.1 Datasets

We perform our analysis across four question answering datasets: TriviaQA, a standard dataset that has been used on a variety of QA related tasks (Joshi et al., 2017); Natural Questions (closed-book) which were aggregated queries to Google (Kwiatkowski et al., 2019a); CountryQA, which we constructed using names of countries and capitals in the form of "What is the capital of Afghanistan?"; Jeopardy questions which were crawled from a fan-created database, J! Archive.[1] We treat this as an open-ended question answering task with a single-token answer by subsampling the datasets down to questions which have a single token answer based on GPT3's vocabulary. This avoids the potential confounder of generating subtokens for OOV answers (which allows us to more accurately measure the probability placed on the gold answers).

We use a random subset of 200 questions for three of our datasets and calculate 95% confidence intervals using bootstrap resampling. For CountryQA, there are 53 questions whose answers were in vocabulary, of which we sample 50. We test each of our fifty templates across this subset of questions

[1] https://j-archive.com/

3.2 LM, Prompting, and Evaluation Details

We study OpenAI's GPT3 model (davinci 175B) as it is a commonly used large language model with reasonable statistical calibration properties (Liang et al., 2022a). For the generated tokens, we take the sum of the probability assigned to the gold answer(s) to be the *probability-on-gold*. When calculating accuracy, we generate 10 tokens and if any of the tokens matches the answer, we'll count that as a correct generation. This is done to not unfairly disadvantage templates that are prone to generating words prior to emitting the answer (e.g., "Allegedly, it's said to be...") (details in A.1).

4 The Impact of Uncertainty on Language Generation

Information in real-life text is rarely in black-and-white, and expressions of uncertainty are necessary in supporting decision-making processes. In this section, we investigate how GPT3's generation changes based on the expressions of uncertainty in its prompt (Figure 2, top) and whether some expressions of uncertainty can have systematic effects on model behavior.

We begin this section by studying two hypotheses for how LMs might respond to expressions of uncertainty. The first hypothesis is that a model "knows what it knows" and regardless of the uncertainty templates used, the answer remains the same. In other words, the model is robust to adding expressions of uncertainty in its input. The second hypothesis is that models will respond differently based on the uncertainty cues, and ideally in a calibrated manner. Under this hypothesis, a confident template would be more likely to produce the correct response than a low confidence template. This hypothesis would be consistent with prior work which has shown that LMs can generate language in the style of diverse personas (Lee et al., 2022; Park et al., 2022).

4.1 Variation in GPT3 Responses Across Verbal Uncertainties

We evaluate GPT3 using zero-shot prompting on our four datasets and find that our results do not support the first hypothesis: we find that GPT3 is *highly* sensitive to uncertainty cues, and accuracy changes significantly depending on the uncertainty expression used. Across our datasets, we find that accuracies can change by up to 80% on the exact same set of questions. This is especially pronounced in CountryQA where a template like, "We realize it's.." achieves 14% accuracy while many other templates result in perfect accuracy. In TriviaQA, when prompted with a template such as "I'm certain it's..." the model's accuracy is 42% but when prompted with "I would need to double check but maybe it's...", accuracy actually increases to 56%. Our findings illustrate that expressions of uncertainty affect language generation and that the changes resulting from these expressions have substantive impact on overall accuracy.

Returning to our second hypothesis, we seek to understand if GPT3's responses to QA answers are linguistically calibrated based on the expression of uncertainty used in the prompt. Surprisingly, we find that weakeners perform significantly better than strengtheners across all four of our datasets. The average accuracy among weakeners across all four datasets is 47% compared to 40% among strengtheners. This effect is especially large in CountryQA where the accuracy gap is 17%. This effect is driven by the use of factive verbs in strengtheners (as nearly all uses of factive verbs in our templates are strengtheners)[2], and the use of factive verbs consistently results in significant losses in accuracy (Figure 3). In other words, when the template presupposes the truth, accuracy drops.

This finding contradicts the second hypothesis, as we might have expected expressions of certainty to improve performance, not hurt it. This is particularly concerning as confident prompts which intuitively seem like they might result in better generations actually lead to worse generations.

Furthermore, we find that in three of our four datasets, the use of evidential markers significantly improves performance. In fact, some of the best performing templates include evidential markers with a source. The top ten performing prompts for each dataset are listed in Appendix C.

The results across the other linguistic features are mixed. Across the four datasets, there is not a consistent improvement from the use of plausibility shields, sources, or personal pronouns. (Figure 7).

4.2 A Redistribution of Probability Mass When Prompted with Weakeners

What explains our surprising finding that templates with weakeners outperform templates with strengtheners? One hypothesis could be that weakeners are changing the underlying probability distribution of

[2] The exception being "I vaguely remember it's...".

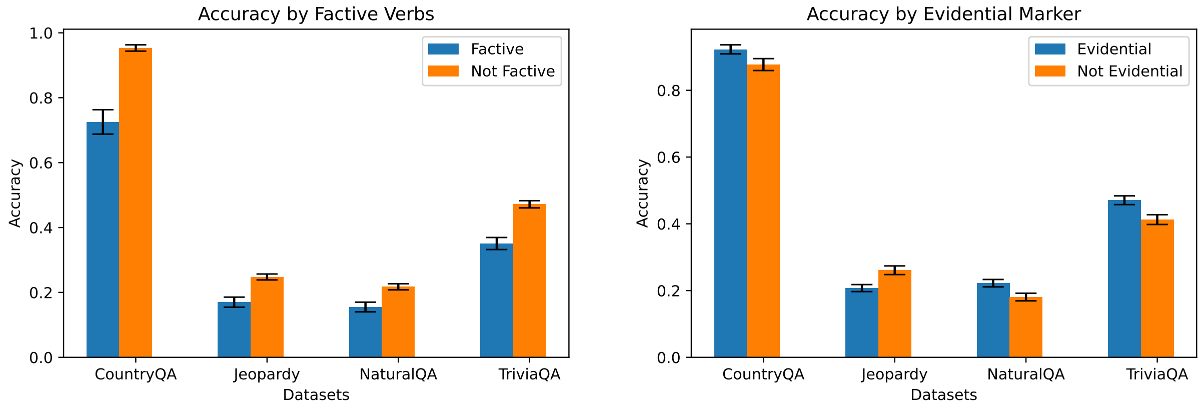


Figure 3: Significant and consistent accuracy losses for templates with factive verbs (left). Use of evidential markers significantly improves accuracy in three out of our four datasets (right). 95% CI calculated using bootstrap resampling.

the potential answers, placing more weight on the correct generations.

If weakeners change the answer distribution, we might expect weakeners to induce an increase in the probability-on-gold (which is defined as the sum of the probabilities placed on all of the answer aliases). We calculate the average probability-on-gold among all correctly predicted answers and find this is not the case. In fact, the probability-on-gold from templates with weakeners is slightly lower than the probability-on-gold from templates with strengtheners. This is true across three of our four datasets NaturalQA (42% vs 45%), JeopardyQA (47% vs 51%), and TriviaQA (53% vs 55%).

Furthermore, we find that weakeners led to a flattening of the distribution of probability mass across answers, compared to strengtheners. We look at the entropy of the probability distribution of top tokens not counting the top prediction; essentially the uncertainty among all but the top candidate. This entropy is significantly higher among weakeners than strengtheners (Table 1). Our finding suggests that the increase in accuracy of weakeners is not due to an increase in answer confidence, but rather when a weakener is used, the model responds by placing probability more evenly across each of the remaining possible options.

4.3 Expressions of Uncertainty Compared to the Standard Prompting Method

Lastly, we find that in certain cases, the use of expressions of uncertainty might actually lead to better performance than the standard prompting method (i.e., just simply using "Q: <question> A:").

Dataset	weakeners	strengtheners
TriviaQA	**2.980** ± 0.01	2.917 ± 0.01
CountryQA	**3.078** ± 0.02	2.875 ± 0.03
Jeopardy	**3.170** ± 0.01	3.089 ± 0.01
NaturalQA	**3.167** ± 0.01	3.106 ± 0.01

Table 1: Average entropy of the probability distribution of alternative tokens among weakeners and strengtheners. Across all four datasets, entropy is higher among weakeners, an indication the model places probability more evenly across the alternative answers. 95% CI calculated using standard error.

In TriviaQA and the template "Online says it's..." achieves an accuracy of 66% compared to 63% achieved by the standard method. In Natural Questions, there are seven templates that outperform the standard method, six of which are expressions of uncertainty. Although these results vary across datasets, we see promising results suggesting that including uncertainty may not only help human decision makers, it may also improve the absolute accuracy of the model.

4.4 The Impact of the Degree of Uncertainty on Performance

Results from Section 4.1 illustrate that GPT3's generation is highly sensitive to uncertainty in prompts, and certainty seems to lead to diminished accuracy. Here, we extend these results to study uncertainty at a more fine-grained level, allowing us to ask if the *degree of uncertainty* could play a role in the accuracy of model generation.

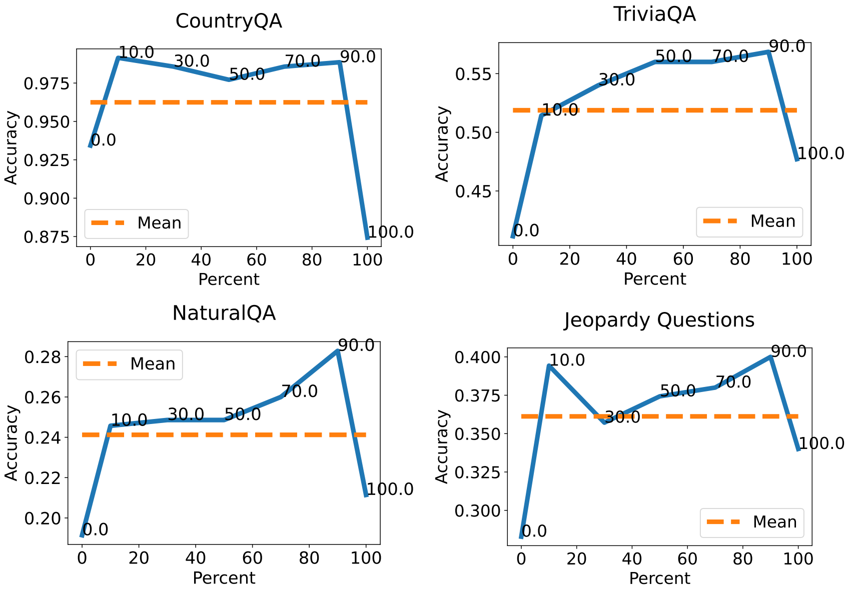


Figure 4: The X-axis indicates the percentage that was injected into the verbal uncertainty. The Y-axis indicates the accuracy across numerical uncertainties. Note the consistent drop in accuracy between 90% and 100% uncertainty and the increase in accuracy between 0% and 10% uncertainty.

Introducing Numerical Values To study the role of uncertainty at a more fine-grained level rather than just certain (strengthener/factive) and uncertain (weakener), we introduce numerical values into our verbal expressions of uncertainty. The setup of our task changes from "What is the capital of Belarus? *I'm sure it's...*" to "What is the capital of Belarus? *I'm 90% sure it's...*". We use a set of seven expressions covering a range of numerical uncertainty expressions, including those that use personal pronouns to weaken uncertainty ("I'm 90% certain...") and those which indicate uncertainty probabilistically but without mentioning the self ("70% chance it's..."). We also downsample our test set to 50 questions per dataset and evaluate each template at 0%, 10%, 30%, 50%, 70%, 90% and 100% intervals.

4.4.1 100% Certainty is not 100% Accurate

Our findings for numerical uncertainties extend our earlier analysis by enabling us to obtain more fine-grained results on whether a model is linguistically calibrated, and how much certainty causes performance degradation.

First, we find that numerical uncertainties are poorly calibrated. For example, the prompt "I'm 90% sure it's..." in TriviaQA only produces the correct answer 57% of the time. Formally, we evaluate the expected calibration error (ECE) and find poor values ranging from 0.50 to 0.30, 0 being the best (Figures 4).

Second, consistent with our findings from Section 4, we find that certainty does hurt model performance. However, we find this is true only at the extremes. We see performance on models peaks usually between 70% and 90% in numerical values but drops in accuracy when using 100% in the prompt. Across all four datasets, with seven templates each, 21 out of the 28 templates which use 100% numerical values had a lower probability-on-gold than templates which used 90% numerical values (Figure 8). Additionally, at the other extreme, when 0% is used in templates, there is also a drastic drop to accuracy.

The Effect of Hyperbolic Language What could be causing these results? We hypothesize that this could be caused by the use of hyperbolic or exaggerated language in the training set, in which numbers are used non-literally. When someone says "I'm 100% certain there was pie left", they don't necessarily mean they are 100% certain is there pie — but rather are speaking loosely to **emphasize** their strong belief there was pie. Similarly, saying "I have zero confidence my car will start" isn't a literally calculated value but just an expression of doubt in their car's abilities. We hypothesize that models somehow recognize the idiomatic, non-literal uses of these extreme values, resulting in lower performances on tasks when introduced into prompts.

Another confounder in how model's interpret numerical values could be the distribution of numerical frequencies in typical model training data. Querying the Pile (Gao et al., 2020), a popular dataset used to train GPT-like models, we find that there are drastic imbalances in the use of percentages in training datasets. There are significant spikes in frequency at the upper extremes (50%, 95% and 100%) (Figure 5). This might be happening as humans might naturally exaggerate values or use colloquial terms to describe confidence. The presence of scientific language like "95% confidence interval" could be another possible source of imbalance. Although spoken natural language also includes rounded percentages, the use of only textual data might be further exacerbating this bias.

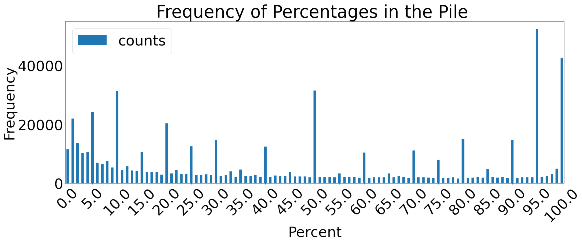


Figure 5: Visualization of the Frequency of percentages found in the pile. Note the peaks at the extremes, (0, 50, 10), and peaks at every 10 and 5 intervals from the first 1,000,000 samples queried from the Pile dataset using the HuggingFace API.

5 When LMs Emit Their Own Uncertainty

Having studied the impact of uncertainty as part of the prompt, we turn to our second motivating question and ask: how does model performance change when models learn to emit their own expressions of uncertainty? This question is becoming increasingly important as researchers have studied training models to generate expressions of uncertainty (Lin et al., 2022; Kadavath et al., 2022; Mielke et al., 2022). Here, we study how model performance changes based on in-context learning examples. Specifically, we follow Lin et al. (2022)'s method in few-shot learning with 50 samples which has been shown to be nearly as effective as fine-tuning on datasets that are magnitudes larger. To ensure that our in-context learning dataset covers a range of confidence levels, our dataset contains 48 samples whose probability-on-gold is uniformly distributed (in buckets of 10) between 0 and 100.

5.1 Experiment Details

To study how LMs respond when emitting their own uncertainty, we follow Lin et al. (2022)'s setup but modify it for the strengtheners and weakeners, which are inherently non-numerical (Figure 2). In our setting, instead of teaching a model to output a percentage confidence, we teach it to output a strengthener when the confidence is above a threshold and nothing otherwise.[3] Conversely, when we study weakeners, we teach it to output a weakener when the probability is below a threshold and nothing otherwise.

As an example, consider the question "What is the capital of France". We record the LM's probability over Paris (the probability-on-gold) and append "*I'm sure*" to the in-context example if the model's confidence was above 0.5. We repeat this for all the in-context examples to obtain our in-context learning training set.

5.2 Prompting Perturbations

Recent work has shown the drastic differences that appear based on simple changes to prompting setup (Suzgun et al., 2022; Lu et al., 2021). We design our in-context learning samples with various perturbations to ensure robustness in our results.

We select high performing weakeners and strengtheners from Section 4 and experiment with appending expressions of uncertainty after the answer (e.g., "Paris. I think") (Table 9).[4] We then use three different sample orderings to perturb our learning samples: ascending and descending order of probability-on-gold and random ordering.[5] Finally, we experiment with a variety of thresholds (0.3, 0.5, 0.7, 0.9) for determining when expressions of (un)certainty should be inserted into the example. These perturbations are done on a small scale to help identify the best hyper-parameters (threshold, placement, and ordering) to use.

We find that varying the threshold across does not drastically change accuracy (with all methods attaining $\sim$ 83% in accuracy), although the threshold does significantly impact the balance of the training datasets. Similarly, we find limited differences in the ordering of the samples. With these

[3] We choose to emit nothing rather than emit an expression of uncertainty, as we wish to isolate the effect of each linguistic expression of uncertainty.

[4] Here, we exclude expressions with attribution shields for concerns of false attribution, more on this in the discussion.

[5] In the ascending and descending orders, all the samples in the beginning or all the samples at the end will include expressions of (un)certainty.

results, we choose a threshold of 0.5 (creating a balanced dataset) and random ordering (simplest setting) as our hyper-parameters for our remaining scenarios and analysis, which we test on 100 TriviaQA questions.

5.3 Results

Gains in Model Calibration When Learning Uncertainty Overall, GPT3 has a limited ability to learn naturalistic expressions of uncertainty in a calibrated manner. We measured calibration based on whether models successfully emit the expressions of uncertainty when the probability of the top token above or below our training threshold. In our setup, when learning to emit certainty, answers with probability-on-gold of greater than 0.5 had a strengthener and answers with less than 0.5 had nothing. Therefore, in its generation when the probability on the top token is greater than 0.5, we'd expect the model to also generate a strengthener and vice versa for weakeners. We measure whether the model successfully generates strengtheners and weakeners through the F1 score, and find that the template with the highest macro-F1 score for uncertainty templates to be 0.56 compared to 0.53 for certainty templates.[6] This is close to the random guessing baseline on this test set which results in an F1 score of 0.45.

To illustrate the difference in calibration between uncertainty and certainty, we can look at the average accuracy when a model emits an expression or not. When learning to express weakeners, the generation of a weakener results in an accuracy of 74% but this increases to 83% when the model doesn't generate a weakener. This is the intended behavior, with the model hedging answers it is more likely to get incorrect. However, when teaching the model to emit strengtheners, the generation of strengtheners does not lead to a significant increase in accuracy (79% with or without an emission of strengtheners). This means that when the model emits certainty, the answer is not more likely to be correct, creating a concerning issue for linguistic calibration (Table 2).

Entropy	**Emitting**	**Not Emitting**
Uncertainty	**0.699***	0.522
Certainty	0.461	**0.617***
Control	N/A	0.541
Accuracy	**Emitting**	**Not Emitting**
Uncertainty	0.738	**0.829***
Certainty	0.799	0.789
Control	N/A	0.78

Table 2: Entropy is higher when uncertainty is being expressed and also higher when certainty is not expressed. Accuracy is also higher when uncertainty is not expressed but accuracy is not significant higher when certainty is expressed (an indication or poor calibration). *Significantly higher value calculated using two-sample t-test, $p < 0.05$.

Modeling Changes in Entropy Despite low model calibration when emitting expressions of certainty and uncertainty, we find that the underlying entropy of the probability distribution of the generated answer is well-calibrated to the expressions of uncertainty. Analyzing the top five predictions for each token, we find that when teaching models weakeners, the entropy of the distribution of potential generations is higher when a weakener is emitted and lower when it is not. The inverse is true when teaching models strengtheners, entropy is lower when strengtheners are emitted and higher when it is not. Although the calibration scores are not strong for either uncertainty or certainty, we see promising behaviors in the entropy of the model's top generations. When emitting weakeners, the model places more consideration on alternative answers and less when emitting strengtheners.

Sensitivity to Placement of Template Finally, we test how model performance differs based on the placement of the templates. The simple design difference of probing for the answer before (e.g., "I think it's Paris.") or after (e.g., "Paris. I think.") an expression of uncertainty can have a significant difference in performance. In our tables, we refer to these places as prefixes (before) and suffixes (after). We find that when appending expressions of uncertainty as a prefix, the generation is significantly worse for accuracy (63% vs 80%). This is also correlated with probability-on-gold being lower in prefixed templates (40% vs 67%). An explanation for this might be that the probability of generating the correct answer will be lower if generated after a phrase like "I think it's..." rather than just generated immediately after the question. Our work suggests that ordering effects may be important when addressing accuracy-calibration

[6] Average F1 scores being .52 average for uncertainty and .49 average of certainty

trade-offs in LMs and that there are accuracy gains when prompting the model to respond with answers as soon as possible.

Template	Certainty	Prob	Top 1
Prefix	Uncertain	0.388	0.592
	Certain	0.407	0.674
Suffix	Uncertain	**0.674**	**0.800**
	Certain	0.673	0.792
Control	N/A	0.673	0.780

Table 3: Average probability on generated token and top 1 accuracy across prefix and suffix templates.

6 Related Work

While scholars have studied model uncertainty, prior work has focused on more accurately extracting model confidence (Kuhn et al., 2023; Sun et al., 2022; Gleave and Irving, 2022), measuring (Kwiatkowski et al., 2019b; Radford et al., 2019; Liang et al., 2022b) and improving model calibration (Jiang et al., 2021; Desai and Durrett, 2020a; Jagannatha and Yu, 2020; Kamath et al., 2020; Kong et al., 2020). However, the community has found mixed results on the calibration of neural model (Minderer et al., 2021; Carrell et al., 2022); for example, Desai and Durrett (2020b) shows that pre-trained transformers are relatively well-calibrated meanwhile Wang et al. (2020) found severe mis-calibration in neural machine translation. Another line of work also explore the trade-off between model performance and calibration (Stengel-Eskin and Van Durme, 2022).

Closest to our work, Mielke et al. (2022) propose solutions to reducing model overconfidence through linguistic calibration, Kadavath et al. (2022) experiment with models' ability emit self-confidence after finding that models are relatively well-calibrated, and Lin et al. (2022) teach models to be linguistically calibrated when answering math questions. We build on these works by incorporating other aspects of naturalistic expressions of uncertainty such as evidentials and factive verbs into the analysis of NLG systems.

In addition, semanticists and computational linguists have long studied speaker commitment factors such as factivity (Karttunen, 1971; Degen and Tonhauser, 2022) and projection (Simons et al., 2010), and more recent work include corpora like the CommitmentBank (De Marneffe et al., 2019) which offers naturally occurring examples, as well as new experimental paradigms to investigate speaker commitment (Degen et al., 2019). A wide variety of scholars have examined computational issues in factuality, veridicality, and commitment (Saurí and Pustejovsky, 2009; de Marneffe et al., 2012; Stanovsky et al., 2017; Rudinger et al., 2018; Jiang and de Marneffe, 2021, inter alia) as well as bias (Pryzant et al., 2020; Patel and Pavlick, 2021) and specific devices like hedges (Prokofieva and Hirschberg, 2014; Raphalen et al., 2022), and modality (Pyatkin et al., 2021).

We see our work as a bridge between these two areas of speaker commitment and natural language generation, by telling us how models interpret and generate speaker commitment through expressions of certainty and uncertainty.

7 Discussion and Conclusion

In this work, we analyzed how naturalistic expressions of uncertainty impact model behavior both for prompting and in-context learning. We find a drop in accuracy when naturalistic expressions of certainty (i.e., strengtheners and factive verbs) are used in zero-shot prompting; this is also true with numerical uncertainty when idioms like "I'm 100% certain" are used. We also find calibration gains when teaching models to express weakeners rather than strengtheners. Moving forward, we present a number of recommendations for the community for NLG and naturalistic expressions of uncertainty.

A Shift to Uncertainty Beyond calibration, teaching models to only emit expressions of uncertainty when they are unsure, rather than when they are sure, could be a safer design choice for human-computer interactions. Prior work has shown cases where AI-assisted decision-making performed worse than human decision-making alone, suggesting an over-reliance on AI, even when systems are wrong (Jacobs et al., 2021; Bussone et al., 2015). This leads us to suspect that teaching models to emit expressions of certainty could further exacerbate these challenges — especially given how poorly calibrated and brittle the models are. Current literature has focused on emitting expressions of confidence in a calibrated manner, but what remains to be investigated is how humans interpret generated naturalistic expressions. As a recommendation, we urge the community to focus on training models to emit expressions of

uncertainty meanwhile further work is conducted.

The Limitations of Written Text The analysis of how GPT3 uses expressions of uncertainty highlights another limitation of using just written text for training language generation. We suspect that the use of weakeners in spoken language will vary greatly from that of written language and that cues such as vocal disfluencies could also impact a model's interpretation and generation of uncertainty. Future work could explore the interaction of LMs with spoken expressions of uncertainty.

Verified Attribution We encourage the community to explore how to integrate attributions of uncertainty in a verified manner. In our experiments, we chose not to experiment with generating attributions as we could not guarantee their validity. However, some of the best-performing templates from Section 4 include phrases like "Wikipedia says...". We hypothesize there is potential to attribute knowledge in a way that supports model transparency and encourages user trust and thinking. However, we would caution the community away from arbitrarily generating attributions as it could provide downstream users with a false, yet highly believable attribution.

Using Expressions of Uncertainty to Sow Doubt Lastly, we warn the community about the potential dangers of using expressions of uncertainty maliciously. Imagine a model arbitrarily responding with weakeners when asked to answer controversial topics (e.g., "Well, supposedly it's true..." or "Wikipedia claims it's happening..."). Beyond the helpful ways in which expressions of uncertainty can support human decision-making, expressions of harm can be used by LMs to undermine individuals and to cast doubt on controversial or subjective questions.

As the community expands the capabilities of NLG to general language that is more natural, it is critical we prepare for the opportunity and harms that may arise from naturalistic expressions of uncertainty.

Acknowledgements

Thank you so much to Tolúlọpẹ́ Ògúnrẹ̀mí, Yiwei Luo, Myra Cheng, Rishi Bommasani, and Omar Shaikh for their helpful feedback and advice!

References

Alexandra Y Aikhenvald. 2004. *Evidentiality*. OUP Oxford.

Adrian Bussone, Simone Stumpf, and Dympna O'Sullivan. 2015. The role of explanations on trust and reliance in clinical decision support systems. In *2015 International Conference on Healthcare Informatics*, pages 160–169.

Annabelle Carrell, Neil Mallinar, James Lucas, and Preetum Nakkiran. 2022. The calibration generalization gap. *arXiv preprint arXiv:2210.01964*.

Marie-Catherine de Marneffe, Christopher D. Manning, and Christopher Potts. 2012. Did it happen? the pragmatic complexity of veridicality assessment. *Computational Linguistics*, 38(2):301–333.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124.

Judith Degen and Judith Tonhauser. 2022. Are there factive predicates? an empirical investigation. *Language*, 98(3):552–591.

Judith Degen, Andreas Trotzke, Gregory Scontras, Eva Wittenberg, and Noah D Goodman. 2019. Definitely, maybe: A new experimental paradigm for investigating the pragmatics of evidential devices across languages. *Journal of Pragmatics*, 140:33–48.

Shrey Desai and Greg Durrett. 2020a. Calibration of pre-trained transformers. *arXiv preprint arXiv:2003.07892*.

Shrey Desai and Greg Durrett. 2020b. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, Online. Association for Computational Linguistics.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Adam Gleave and Geoffrey Irving. 2022. Uncertainty estimation for language reward models. *arXiv preprint arXiv:2203.07472*.

Ken Hyland. 2005. Stance and engagement: A model of interaction in academic discourse. *Discourse studies*, 7(2):173–192.

Ken Hyland. 2014. Disciplinary discourses: Writer stance in research articles. In *Writing: Texts, processes and practices*, pages 99–121. Routledge.

Maia Jacobs, Melanie F Pradier, Thomas H McCoy Jr, Roy H Perlis, Finale Doshi-Velez, and Krzysztof Z Gajos. 2021. How machine-learning recommendations influence clinician treatment selections: the example of antidepressant selection. *Translational psychiatry*, 11(1):108.

Abhyuday Jagannatha and Hong Yu. 2020. Calibrating structured output predictors for natural language processing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2078–2092, Online. Association for Computational Linguistics.

Nanjiang Jiang and Marie-Catherine de Marneffe. 2021. He thinks he knows better than the doctors: BERT for event factuality fails on pragmatics. *Transactions of the Association for Computational Linguistics*, 9:1081–1097.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

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

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696, Online. Association for Computational Linguistics.

Lauri Karttunen. 1971. Some observations on factivity. *Research on Language & Social Interaction*, 4(1):55–69.

Paul Kiparsky and Carol Kiparsky. 1970. *FACT*, pages 143–173. De Gruyter Mouton, Berlin, Boston.

Lingkai Kong, Haoming Jiang, Yuchen Zhuang, Jie Lyu, Tuo Zhao, and Chao Zhang. 2020. Calibrated language model fine-tuning for in- and out-of-distribution data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1326–1340, Online. Association for Computational Linguistics.

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019a. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019b. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

George Lakoff. 1975. Hedges: A study in meaning criteria and the logic of fuzzy concepts. In *Contemporary Research in Philosophical Logic and Linguistic Semantics: Proceedings of a Conference Held at the University of Western Ontario, London, Canada*, pages 221–271. Springer.

Young-Jun Lee, Chae-Gyun Lim, Yunsu Choi, Ji-Hui Lm, and Ho-Jin Choi. 2022. PERSONACHATGEN: Generating personalized dialogues using GPT-3. In *Proceedings of the 1st Workshop on Customized Chat Grounding Persona and Knowledge*, pages 29–48, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2022a. Holistic evaluation of language models.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022b. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

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

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Sabrina J Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. 2022. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. 2021. Revisiting the calibration of modern neural networks. *Advances in Neural Information Processing Systems*, 34:15682–15694.

Joon Sung Park, Lindsay Popowski, Carrie Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2022. Social simulacra: Creating populated prototypes for social computing systems. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, pages 1–18.

Roma Patel and Ellie Pavlick. 2021. "was it "stated" or was it "claimed"?: How linguistic bias affects generative language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10080–10095, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

E. Prince, C. Bosk, and J. Frader. 1982. On hedging in physician-physician discourse. *Di Pietro, R.J., Ed., Linguistics and the Professions*, pages 83–97.

Anna Prokofieva and Julia Hirschberg. 2014. Hedging and speaker commitment. In *5th Intl. Workshop on Emotion, Social Signals, Sentiment & Linked Open Data, Reykjavik, Iceland*.

Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. 2020. Automatically neutralizing subjective bias in text. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 480–489.

Valentina Pyatkin, Shoval Sadde, Aynat Rubinstein, Paul Portner, and Reut Tsarfaty. 2021. The possible, the plausible, and the desirable: Event-based modality detection for language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 953–965.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Yann Raphalen, Chloé Clavel, and Justine Cassell. 2022. "You might think about slightly revising the title": Identifying hedges in peer-tutoring interactions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2174, Dublin, Ireland. Association for Computational Linguistics.

Rachel Rudinger, Aaron Steven White, and Benjamin Van Durme. 2018. Neural models of factuality. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 731–744, New Orleans, Louisiana. Association for Computational Linguistics.

Roser Saurí and James Pustejovsky. 2009. Factbank: a corpus annotated with event factuality. *Language resources and evaluation*, 43:227–268.

Mandy Simons, Judith Tonhauser, David Beaver, and Craige Roberts. 2010. What projects and why. In *Semantics and linguistic theory*, volume 20, pages 309–327.

Gabriel Stanovsky, Judith Eckle-Kohler, Yevgeniy Puzikov, Ido Dagan, and Iryna Gurevych. 2017. Integrating deep linguistic features in factuality prediction over unified datasets. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 352–357, Vancouver, Canada. Association for Computational Linguistics.

Elias Stengel-Eskin and Benjamin Van Durme. 2022. Calibrated interpretation: Confidence estimation in semantic parsing. *arXiv preprint arXiv:2211.07443*.

Meiqi Sun, Wilson Yan, Pieter Abbeel, and Igor Mordatch. 2022. Quantifying uncertainty in foundation models via ensembles. In *NeurIPS 2022 Workshop on Robustness in Sequence Modeling*.

Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. 2022. Prompt-and-rerank: A method for zero-shot and few-shot arbitrary textual style transfer with small language models. *arXiv preprint arXiv:2205.11503*.

Shuo Wang, Zhaopeng Tu, Shuming Shi, and Yang Liu. 2020. On the inference calibration of neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3070–3079, Online. Association for Computational Linguistics.

A Appendix

Additional Details

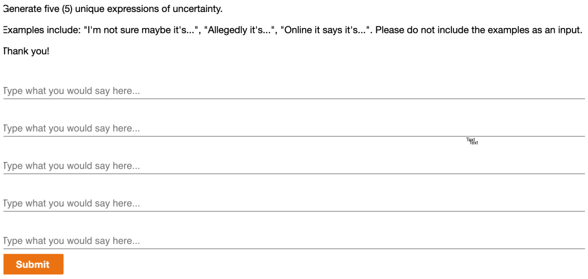
A.1 Additional Details from Section 4

The model generates up to 10 tokens for each question and returns the top predictions per token. For Section 4 we use OpenAI's researcher API and retrieve the top 50 most probable predictions per token. For Section 5 we use the standard API and retrieve the top 5 most probable predictions per token.

We are careful that our prompts do not end with trailing white space (" ") as recommended by OpenAI in order to prompt the best generations. We also use the delimiters "Q:" and "A:" to signal the start and end of questions and answers.[7]

B Amazon Mechanical Turk Results

We use Amazon Mechanical Turk to crowd-source some additional expressions of uncertainty. A screenshot of the task is included below. Workers were filtered to be have HITs greater than 99 and to have at least 500 approved HITs. Given the simplicity of the task, we estimated it would take users a minute or two to complete the task, a paid users $0.35 USD for the task which results in roughly $10.50 USD to $21.00 USD an hour. We collected a total of 9 samples of 5 examples each. The authors then read, filtered, and modified the examples to follow the overall linguistic structure of the other templates.



Generate five (5) unique expressions of uncertainty.

Examples include: "I'm not sure maybe it's...", "Allegedly it's...", "Online it says it's...". Please do not include the examples as an input.

Thank you!

Type what you would say here...

Type what you would say here...

Type what you would say here...

Type what you would say here...

Type what you would say here...

Figure 6: Screenshot of the Crowdsourced Example

C Top 10 Templates from Section 4

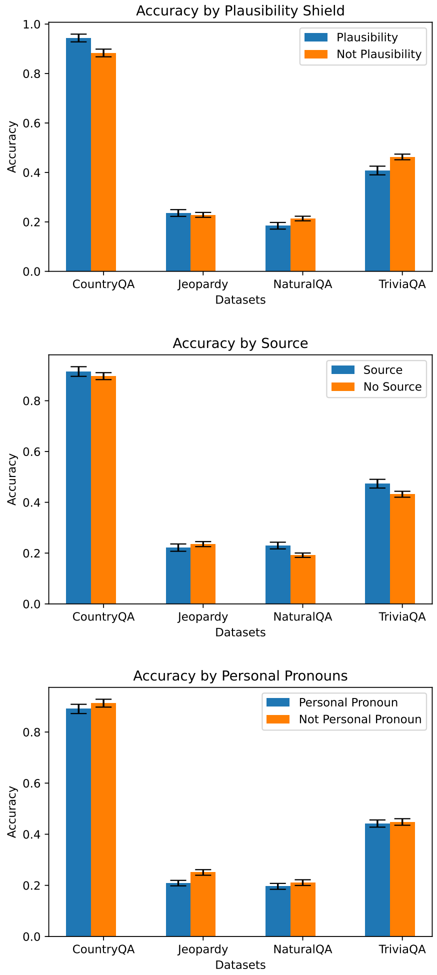


Figure 7: The use of plausibility shields, sources, and personal pronouns are mixed, without significant consistent improvements or drops in accuracy. 95% CI calculated using bootstrap resampling.

[7] In CountryQA and TriviaQA, an additional new line character was added before "A:". To reduce accruing additional costs, NaturalQA and Jeopardy results were not rerun to match this exact template.

Template	Strengtheners	Shield	Evidential Marker	Factive Verb	Source	1P
Apparently it's	Weakener	None	Evidential	Not Factive	No Source	No
Presumably it's	Weakener	None	Evidential	Not Factive	No Source	No
Rumor says it it's	Weakener	None	Evidential	Not Factive	No Source	No
Allegedly it's	Weakener	None	Evidential	Not Factive	No Source	No
I was told it's	Weakener	None	Evidential	Not Factive	No Source	Yes
I've heard it's	Weakener	None	Evidential	Not Factive	No Source	Yes
They told me it's	Weakener	None	Evidential	Not Factive	No Source	Yes
Wikipedia suggests it's	Weakener	None	Evidential	Not Factive	Source	No
Online says it's	Weakener	None	Evidential	Not Factive	Source	No
The internet says it's	Weakener	None	Evidential	Not Factive	Source	No
Wikipedia claims it's	Weakener	None	Evidential	Not Factive	Source	No
Wikipedia says it's	Weakener	None	Evidential	Not Factive	Source	No
I read on the internet it's	Weakener	None	Evidential	Not Factive	Source	Yes
I read on Wikipedia it's	Weakener	None	Evidential	Not Factive	Source	Yes
I read online it's	Weakener	None	Evidential	Not Factive	Source	Yes
To the best of my knowledge it's	Weakener	Plausibility	Evidential	Not Factive	No Source	Yes
As far as I'm aware it's	Weakener	Plausibility	Evidential	Not Factive	No Source	Yes
I vaguely remember it's	Weakener	Plausibility	Evidential	Not Factive	No Source	Yes
It could be	Weakener	Plausibility	Not Evidential	Not Factive	No Source	No
Considering all the options it's	Weakener	Plausibility	Not Evidential	Not Factive	No Source	No
It probably is	Weakener	Plausibility	Not Evidential	Not Factive	No Source	No
Maybe it's	Weakener	Plausibility	Not Evidential	Not Factive	No Source	No
Perhaps it's	Weakener	Plausibility	Not Evidential	Not Factive	No Source	No
It should be	Weakener	Plausibility	Not Evidential	Not Factive	No Source	No
I don't know maybe it's	Weakener	Plausibility	Not Evidential	Not Factive	No Source	Yes
I suppose it's	Weakener	Plausibility	Not Evidential	Not Factive	No Source	Yes
I would need to double check but maybe it's	Weakener	Plausibility	Not Evidential	Not Factive	No Source	Yes
I wouldn't put money on it but maybe it's	Weakener	Plausibility	Not Evidential	Not Factive	No Source	Yes
I'm not an expert but maybe it's	Weakener	Plausibility	Not Evidential	Not Factive	No Source	Yes
I think it's	Weakener	Plausibility	Not Evidential	Not Factive	No Source	Yes
I feel like it should be	Weakener	Plausibility	Not Evidential	Not Factive	No Source	Yes
It is known that it's	Strengthener	None	Evidential	Factive	No Source	No
The most recent evidence shows it's	Strengthener	None	Evidential	Factive	Source	No
The rules state it's	Strengthener	None	Evidential	Factive	Source	No
Two recent studies demonstrate it's	Strengthener	None	Evidential	Factive	Source	No
Wikipedia acknowledges it's	Strengthener	None	Evidential	Factive	Source	No
Wikipedia confirms it's	Strengthener	None	Evidential	Factive	Source	No
Our lab has shown it's	Strengthener	None	Evidential	Factive	Source	Yes
Evidently it's	Strengthener	None	Evidential	Not Factive	No Source	No
According to the latest research it's	Strengthener	None	Evidential	Not Factive	Source	No
We can see in the textbook that it's	Strengthener	None	Evidential	Not Factive	Source	Yes
It must be	Strengthener	None	Not Evidential	Factive	No Source	No
We realize it's	Strengthener	None	Not Evidential	Factive	No Source	Yes
We understand it's	Strengthener	None	Not Evidential	Factive	No Source	Yes
We know it's	Strengthener	None	Not Evidential	Factive	No Source	Yes
Undoubtedly it's	Strengthener	None	Not Evidential	Not Factive	No Source	No
With 100% confidence it's	Strengthener	None	Not Evidential	Not Factive	No Source	No
I'm certain it's	Strengthener	None	Not Evidential	Not Factive	No Source	Yes
I am 100% sure it's	Strengthener	None	Not Evidential	Not Factive	No Source	Yes
It's	None	None	Not Evidential	Not Factive	No Source	No

Table 4: Full list of expressions of uncertainty coded for six linguistic features. *Claims is a neg-factive but in our schema, will just be considered not a factive verb. (Saurí and Pustejovsky, 2009)

	Template	Top 1 Accuracy
0	Online says it’s	0.660
1	Standard Method	0.625
2	Wikipedia confirms it’s	0.600
3	Wikipedia suggests it’s	0.595
4	The internet says it’s	0.585
5	Wikipedia claims it’s	0.575
6	Wikipedia says it’s	0.575
7	We can see in the textbook that it’s	0.565
8	I would need to double check but maybe it’s	0.555
9	Rumor says it it’s	0.550

Table 5: Top 10 Templates For TriviaQA

	Template	Top 1 Accuracy
0	Standard Method	1.0
1	I read on Wikipedia it’s	1.0
2	It’s	1.0
3	It should be	1.0
4	Allegedly it’s	1.0
5	I’m not an expert but maybe it’s	1.0
6	I wouldn’t put money on it but maybe it’s	1.0
7	Presumably it’s	1.0
8	I read online it’s	1.0
9	I read on the internet it’s	1.0

Table 6: Top 10 Templates For CountryQA

	Template	Top 1 Accuracy
0	Standard Method	0.450
1	It must be	0.390
2	It’s	0.380
3	It could be	0.370
4	The internet says it’s	0.370
5	Online says it’s	0.360
6	With 100% confidence it’s	0.350
7	Undoubtedly it’s	0.345
8	Wikipedia says it’s	0.345
9	Wikipedia confirms it’s	0.325

Table 7: Top 10 Templates for Jeopardy

	Template	Top 1 Accuracy
0	Wikipedia claims it's	0.340
1	Wikipedia says it's	0.335
2	Online says it's	0.310
3	Wikipedia suggests it's	0.305
4	The internet says it's	0.300
5	Wikipedia confirms it's	0.300
6	I read on Wikipedia it's	0.295
7	Presumably it's	0.275
8	Standard Method	0.275
9	I think it's	0.270

Table 8: Top 10 Templates for NaturalQA

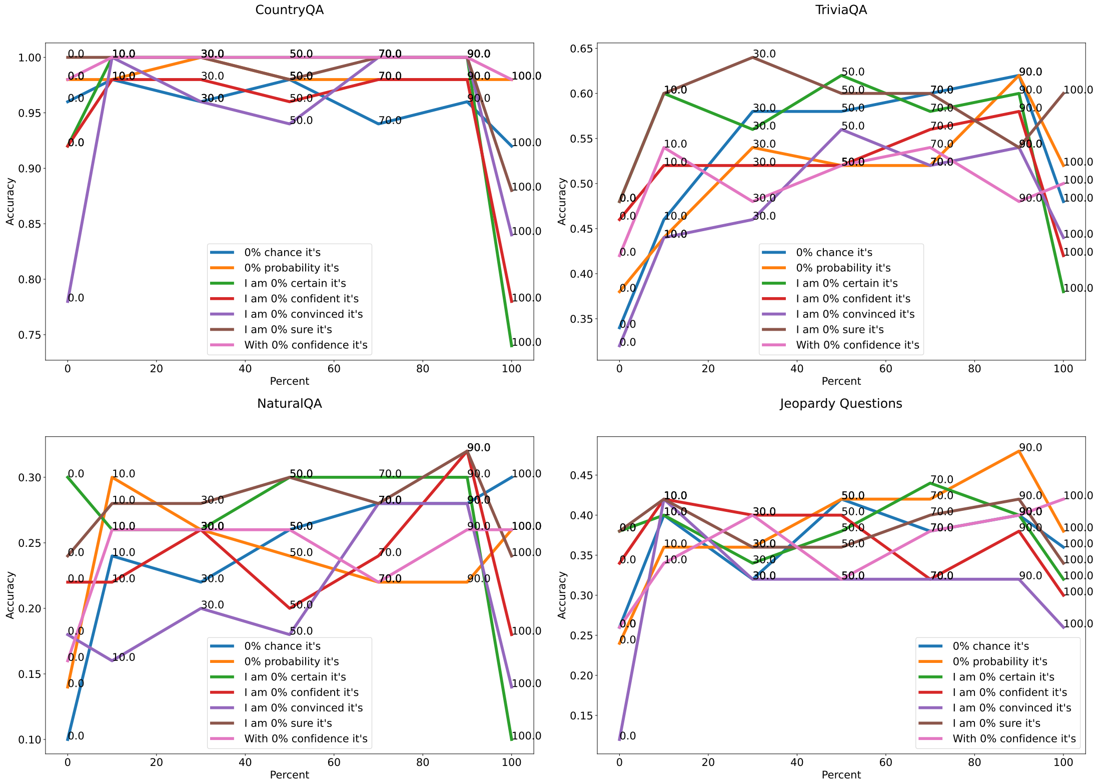


Figure 8: Variation in probability-on-gold across numerical uncertainties. Note the consistent drop in accuracy between 90% and 100% uncertainty and the increase in accuracy between 0% and 10% uncertainty.

Expression	Suffix	Prefix
Uncertainty	Undoubtedly.	Undoubtedly it’s
Uncertainty	With 100% confidence.	With 100% confidence it’s
Uncertainty	We know it.	We know it’s
Uncertainty	Evidently.	Evidently it’s
Uncertainty	It must be.	It must be
Certainty	I think.	I think it’s
Certainty	It could be.	It could be
Certainty	But I would need to double check.	I would need to double check but maybe it’s
Certainty	I suppose.	I suppose it’s
Certainty	But I wouldn’t put money on it.	I wouldn’t put money on it but maybe it’s

Table 9: List of Templates Used for Section 5