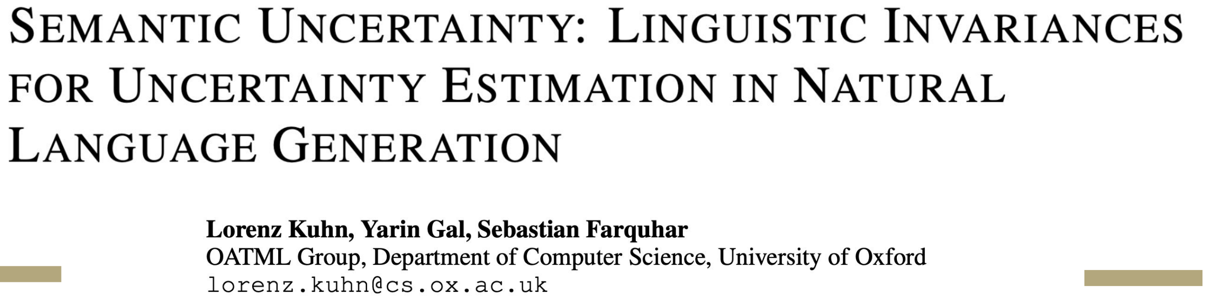


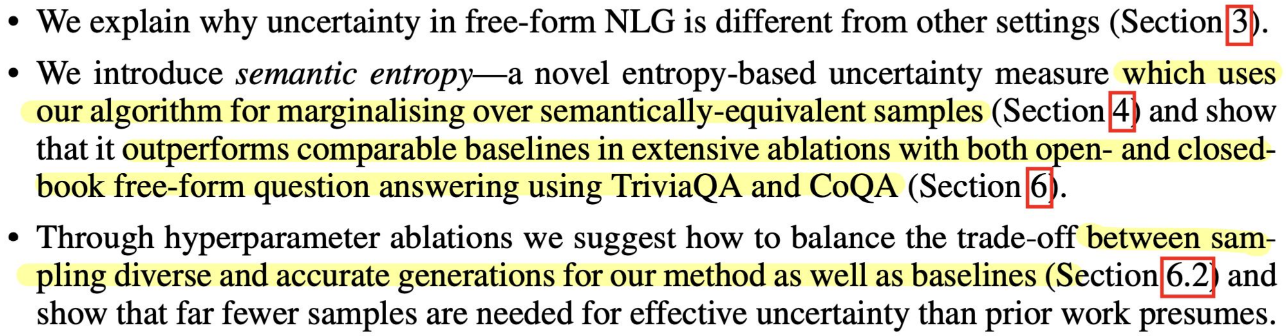
Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation

Lorenz Kuhn, Yarin Gal, Sebastian Farquhar
OATML Group, Department of Computer Science, University of Oxford
lorenz.kuhn@cs.ox.ac.uk

Presentation by: Shrey Pandit

Broad Aim

- We want to know if we can trust the outputs of the LLM
- Natural language has challenge of "Semantic equivalence" - different sentences can mean the same thing
- At token level model is uncertain between the two forms of the same meaning
- We estimate "Semantic likelihood" - probabilities attached to meaning rather than standard sequence



- We explain why uncertainty in free-form NLG is different from other settings (Section 3).
- We introduce *semantic entropy*—a novel entropy-based uncertainty measure which uses our algorithm for marginalising over semantically-equivalent samples (Section 4) and show that it outperforms comparable baselines in extensive ablations with both open- and closed-book free-form question answering using TriviaQA and CoQA (Section 6).
- Through hyperparameter ablations we suggest how to balance the trade-off between sampling diverse and accurate generations for our method as well as baselines (Section 6.2) and show that far fewer samples are needed for effective uncertainty than prior work presumes.

Basics of uncertainty estimation

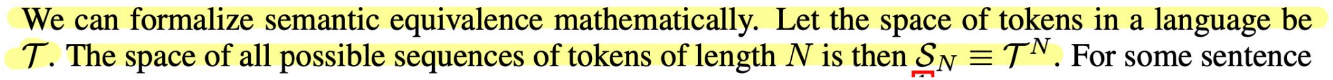
- Total uncertainty of a prediction is the predictive entropy of the output distribution - This measures the information one has about the output given the input
- Entropy is highest when the output is minimally informative, all “classes” have same probability
- Predictive entropy for a point x is conditional entropy of the output random variable Y with realisation y given x

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

Challenges in uncertainty estimation for NLG

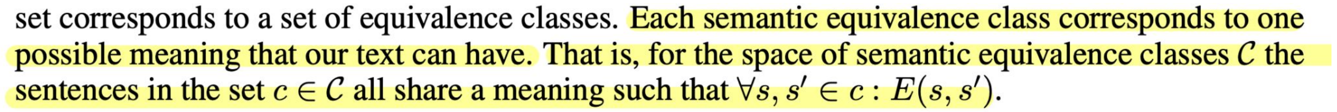
Semantic Equivalence in LLM output

“The capital of France is Paris” means same as “France’s capital is Paris”



We can formalize semantic equivalence mathematically. Let the space of tokens in a language be $\mathcal{T}$. The space of all possible sequences of tokens of length N is then $\mathcal{S}_N \equiv \mathcal{T}^N$. For some sentence

Semantic equivalence relation E(. , .) holds any two sentence that means the same thing



set corresponds to a set of equivalence classes. Each semantic equivalence class corresponds to one possible meaning that our text can have. That is, for the space of semantic equivalence classes $\mathcal{C}$ the sentences in the set $c \in \mathcal{C}$ all share a meaning such that $\forall s, s' \in c : E(s, s')$.

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

Challenges in uncertainty estimation for NLG

Sampling the extremely high dimension language space

Estimating predictive entropy requires taking an expectation in output-space. However, the output-space of natural language has O(|T|^N) dimensions

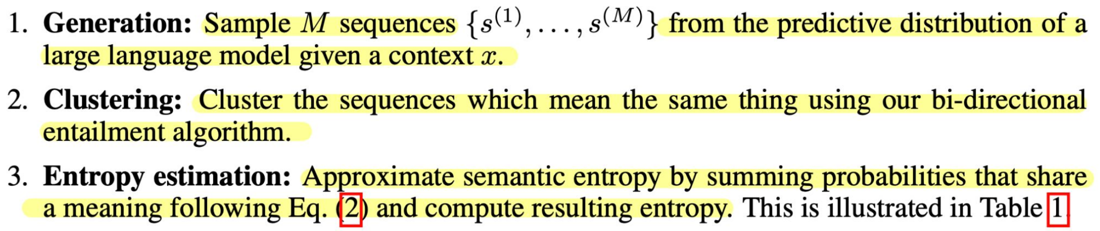
Variable length generations

Longer sentence -> lower joint likelihood (The joint likelihood of a sequence of length N shrinks exponentially in N)

Its negative log-probability therefore grows linearly in N , so longer sentences tend to contribute more to entropy.

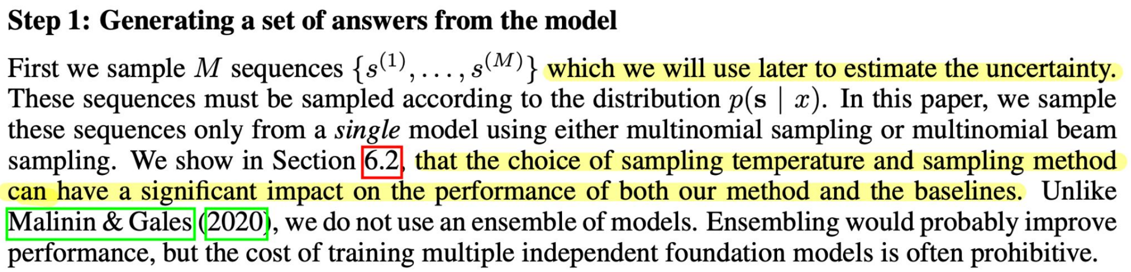
Semantic Uncertainty (Main contribution)

“Uncertainty over meanings is more important for most situations than uncertainty over the exact tokens used to express those meanings”



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. This is illustrated in Table 1.

1. Generation

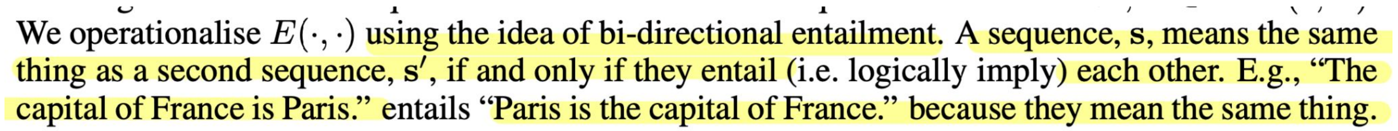


Step 1: Generating a set of answers from the model

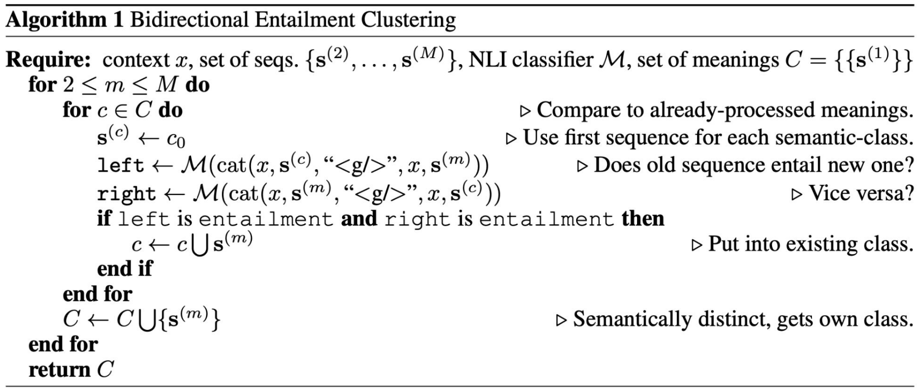
First we sample M sequences $\{s^{(1)}, \ldots, s^{(M)}\}$ which we will use later to estimate the uncertainty. These sequences must be sampled according to the distribution $p(\mathbf{s} \mid x)$. In this paper, we sample these sequences only from a *single* model using either multinomial sampling or multinomial beam sampling. We show in Section 6.2, that the choice of sampling temperature and sampling method can have a significant impact on the performance of both our method and the baselines. Unlike Malinin & Gales (2020), we do not use an ensemble of models. Ensembling would probably improve performance, but the cost of training multiple independent foundation models is often prohibitive.

Semantic Uncertainty (Main contribution)

2. Clustering


We operationalise $E(\cdot,\cdot)$ using the idea of bi-directional entailment. A sequence, $\mathbf{s}$, means the same thing as a second sequence, $\mathbf{s}'$, if and only if they entail (i.e. logically imply) each other. E.g., "The capital of France is Paris." entails "Paris is the capital of France." because they mean the same thing.

Use the common, NLI task to label entailment, neutral, and contradiction


Algorithm 1 Bidirectional Entailment Clustering

```
Require: context $x$, set of seqs. $\{\mathbf{s}^{(2)}, \dots, \mathbf{s}^{(M)}\}$, NLI classifier $\mathcal{M}$, set of meanings $C = \{\{\mathbf{s}^{(1)}\}\}$
  for $2 \le m \le M$ do
      for $c \in C$ do                                                    ▷ Compare to already-processed meanings.
          $\mathbf{s}^{(c)} \leftarrow c_0$                                ▷ Use first sequence for each semantic-class.
          left $\leftarrow \mathcal{M}(\text{cat}(x, \mathbf{s}^{(c)}, \text{"<g/>"}, x, \mathbf{s}^{(m)}))$      ▷ Does old sequence entail new one?
          right $\leftarrow \mathcal{M}(\text{cat}(x, \mathbf{s}^{(m)}, \text{"<g/>"}, x, \mathbf{s}^{(c)}))$     ▷ Vice versa?
          if left is entailment and right is entailment then
              $c \leftarrow c \bigcup \mathbf{s}^{(m)}$                     ▷ Put into existing class.
          end if
      end for
      $C \leftarrow C \bigcup \{\mathbf{s}^{(m)}\}$                        ▷ Semantically distinct, gets own class.
  end for
  return $C$
```

Note: "Paris." does not entail "The capital of France is Paris."

Semantic Uncertainty (Main contribution)

3. Computing the semantic entropy

Add the likelihood of the entire cluster, as a likelihood of each meaning

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

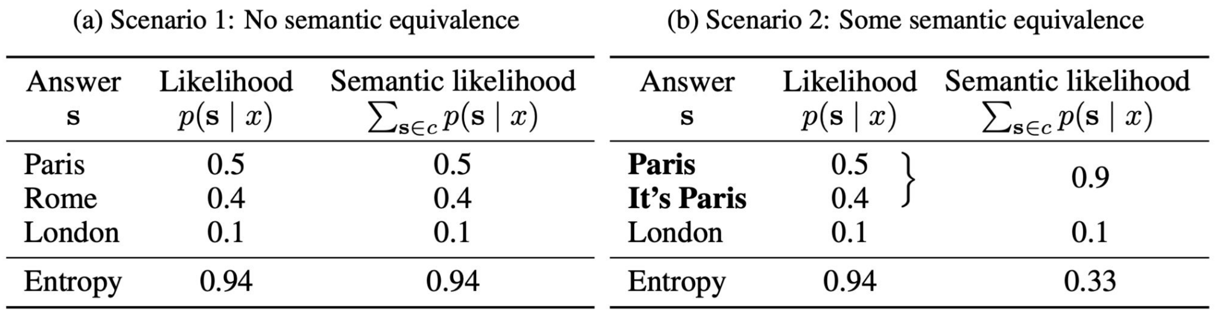
We do not have access to every possible meaning-class c, so we can only sample c from the distribution induced by the model. To handle this, we estimate the expectation using Monte-Carlo integration over semantic equivalence classes C

$$SE(x) \approx -|C|^{-1} \sum_{i=1}^{|C|} \log p(C_i \mid x).$$

Semantic Uncertainty (Main contribution)

How does this semantic entropy address the challenge of NLG

Generations whose meanings are the same but differ on unimportant tokens will be added together, which we expect will reduce the effect of the likelihoods of unimportant tokens although we do not demonstrate this empirically.



(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

Empirical Evaluation

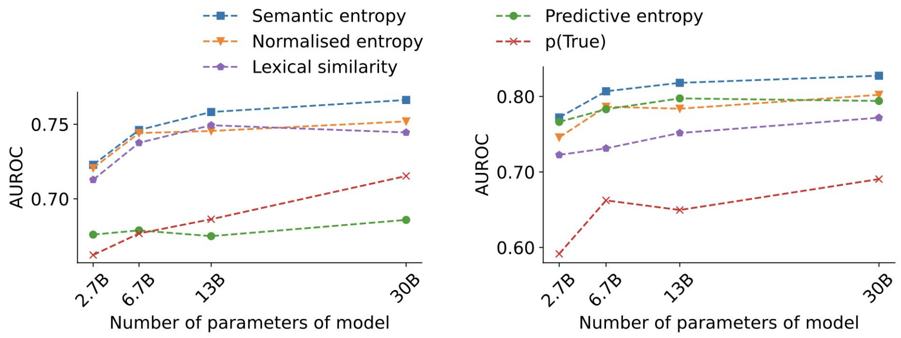
Let's recall why we need to measure uncertainty in LLMs-

It should offer information about how reliable the model’s answers are—that is, very uncertain generations should be less likely to be correct.

Metric - AUROC metric is equivalent to the probability that a randomly chosen correct answer has a higher prediction score than a randomly chosen incorrect answer. Higher scores are better, with perfect uncertainty scoring 1 while a random uncertainty measure would score 0.5

Empirical Evaluation

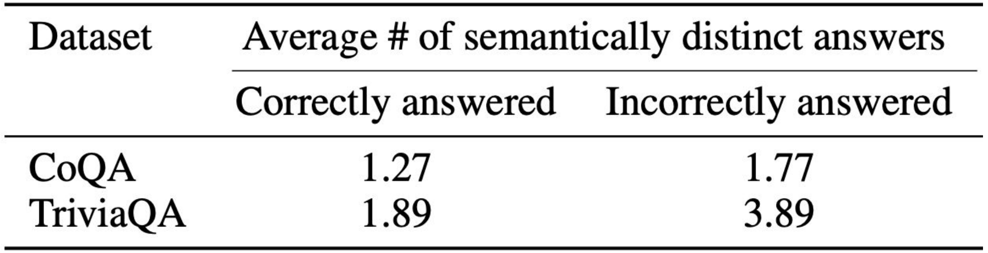
Semantic - Their proposed method
Normalised - Divides the joint log-probability of each sequence by the length of the sequence,
Lexical - Average similarity score in the answer set
Predictive - Standard predicted entropy without length normalization



(a) CoQA (b) TriviaQA

If the model answered a question correctly then higher chance that it is sure, so number of distinct answers (number of clusters C) should be low.

For incorrect answers, it should be higher, which can be seen in the table



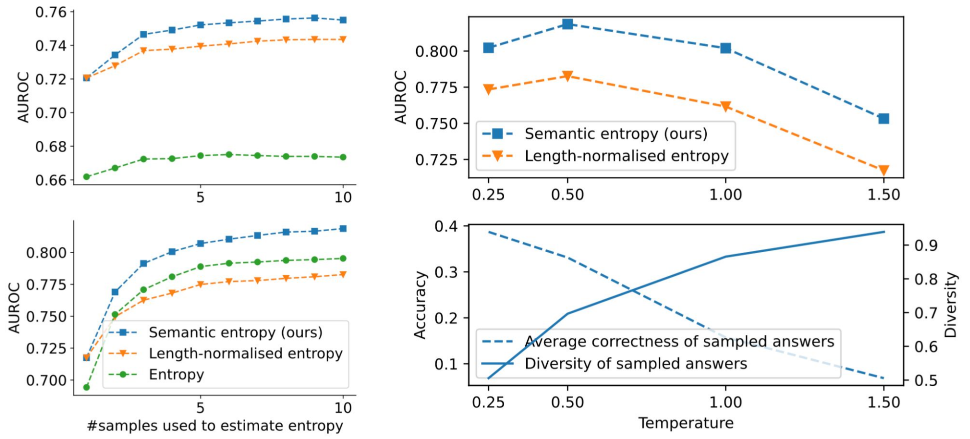
Dataset	Average # of semantically distinct answers	
	Correctly answered	Incorrectly answered
CoQA	1.27	1.77
TriviaQA	1.89	3.89

Hyperparameter for effective sampling

Increasing the temperature increases the diversity of samples, expect more diverse generations to cover the space of possible meanings more fully

Reducing the temperature improves the average correctness of the answer, more accurate models are also better at estimating uncertainty.

These two effects compete and the highest AUROC for semantic entropy and length-normalised entropy is optimised by an intermediate temperature of 0.5. A lower temperature would improve accuracy, while a higher temperature would improve diversity



Conclusion

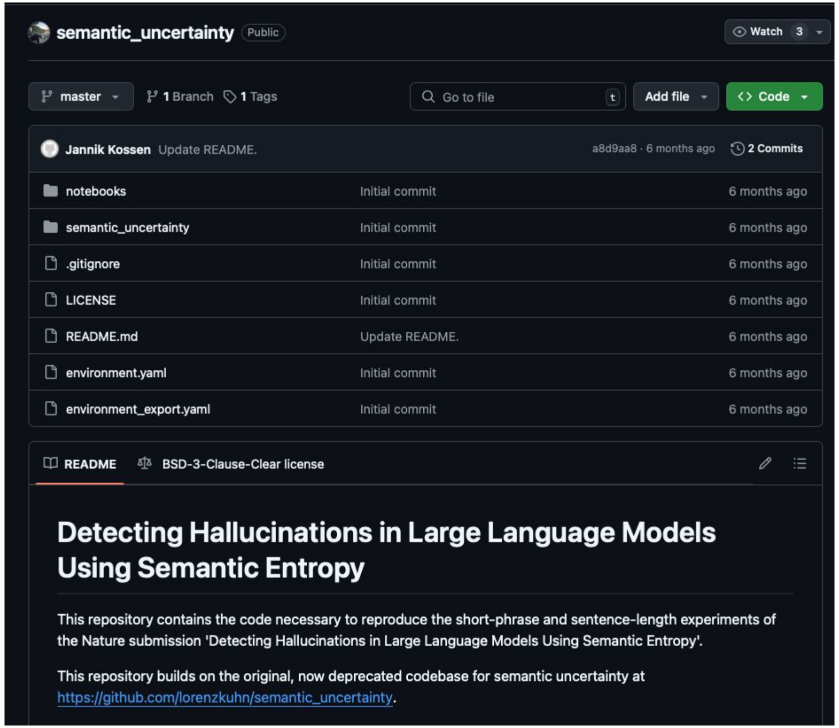
Github Link - https://github.com/lorenzkuhn/semantic_uncertainty

Puts forward the issues at free-form NLG entropy calculation

Proposes:

- the entropy of the distribution over meanings rather than sequences
- we introduce a novel bidirectional entailment clustering algorithm which uses a smaller natural language inference model

Shows the effects of temperature, sampling more semantic equivalent sentences on entropy, the proposed method consistently performs better than the other calculation methods



Jannik Kossen Update README. a8d9aa8 · 6 months ago 2 Commits

notebooks	Initial commit	6 months ago
semantic_uncertainty	Initial commit	6 months ago
.gitignore	Initial commit	6 months ago
LICENSE	Initial commit	6 months ago
README.md	Update README.	6 months ago
environment.yaml	Initial commit	6 months ago
environment_export.yaml	Initial commit	6 months ago

README BSD-3-Clause-Clear license

Detecting Hallucinations in Large Language Models Using Semantic Entropy

This repository contains the code necessary to reproduce the short-phrase and sentence-length experiments of the Nature submission 'Detecting Hallucinations in Large Language Models Using Semantic Entropy'.

This repository builds on the original, now deprecated codebase for semantic uncertainty at https://github.com/lorenzkuhn/semantic_uncertainty.