

Fact-Checking the Output of Large Language Models via Token-Level Uncertainty Quantification

Ekaterina Fadeeva^{3,4} \diamond Aleksandr Rubashevskii^{1,3} \diamond Artem Shelmanov¹ \diamond
Sergey Petrakov³ Haonan Li¹ Hamdy Mubarak⁷ Evgenii Tsymbalov⁸ Gleb Kuzmin^{2,5}
Alexander Panchenko^{2,3} Timothy Baldwin^{1,6} Preslav Nakov¹ Maxim Panov¹
¹MBZUAI ²AIRI ³Center for Artificial Intelligence Technology ⁴HSE University
⁵FRC CSC RAS ⁶The University of Melbourne ⁷QCRI ⁸Independent Researcher
{ekaterina.fadeeva, sergey.petrakov}@skol.tech {kuzmin, panchenko}@airi.net
{aleksandr.rubashevskii, artem.shelmanov, haonan.li}@mbzuai.ac.ae
{maxim.panov, timothy.baldwin, preslav.nakov}@mbzuai.ac.ae hmubarak@hbku.edu.qa

Abstract

Large language models (LLMs) are notorious for hallucinating, i.e., producing erroneous claims in their output. Such hallucinations can be dangerous, as occasional factual inaccuracies in the generated text might be obscured by the rest of the output being generally factually correct, making it extremely hard for the users to spot them. Current services that leverage LLMs usually do not provide any means for detecting unreliable generations. Here, we aim to bridge this gap. In particular, we propose a novel fact-checking and hallucination detection pipeline based on token-level uncertainty quantification. Uncertainty scores leverage information encapsulated in the output of a neural network or its layers to detect unreliable predictions, and we show that they can be used to fact-check the atomic claims in the LLM output. Moreover, we present a novel token-level uncertainty quantification method that removes the impact of uncertainty about what claim to generate on the current step and what surface form to use. Our method Claim Conditioned Probability (CCP) measures only the uncertainty of a particular claim value expressed by the model. Experiments on the task of biography generation demonstrate strong improvements for CCP compared to the baselines for seven LLMs and four languages. Human evaluation reveals that the fact-checking pipeline based on uncertainty quantification is competitive with a fact-checking tool that leverages external knowledge.

1 Introduction

Large language models (LLMs) have become a ubiquitous and versatile tool for addressing a variety of natural language processing (NLP) tasks. People use these models for tasks including information search (Sun et al., 2023b), to ask medical

questions (Thirunavukarasu et al., 2023), or to generate new content (Sun et al., 2023a). Recently, there has been a notable shift in user behavior, indicating an increasing reliance on and trust in LLMs as primary information sources, often surpassing traditional channels. However, a significant challenge with the spread of these models is their tendency to produce “hallucinations”, i.e., factually incorrect generations that contain misleading information (Bang et al., 2023; Dale et al., 2023). This is a side-effect of the way modern LLMs are designed and trained (Kalai and Vempala, 2023).

LLM hallucinations are a major concern because the deceptive content at the surface level can be highly coherent and persuasive. Common examples include the creation of fictitious biographies or the assertion of unfounded claims. The danger is that a few occasional false claims might be easily obscured by a large number of factual statements, making it extremely hard for people to spot them. As hallucinations in LLM outputs are hard to eliminate completely, users of such systems could be informed via highlighting some potential caveats in the text, and this is where our approach can help.

Fact-checking is a research direction that addresses this problem. It is usually approached using complex systems that leverage external knowledge sources (Guo et al., 2022; Nakov et al., 2021; Wadden et al., 2020). This introduces problems related to the incomplete nature of such sources and notable overhead in terms of storing the knowledge. We argue that information about whether a generation is a hallucination is encapsulated in the model output itself, and can be extracted using uncertainty quantification (UQ) (Gal et al., 2016; Kotelevskii et al., 2022; Vazhentsev et al., 2022, 2023a). This avoids implementing complex and expensive fact-checking systems that require additional computational overhead and rely on external resources.

Prior work has mainly focused on quantifica-

\diamond Equal contribution

tion of uncertainty for the whole generated text and been mostly limited to tasks such as machine translation (Malinin and Gales, 2020), question answering (Kuhn et al., 2023), and text summarization (van der Poel et al., 2022). However, the need for an uncertainty score for only a part of the generation substantially complicates the problem. We approach it by leveraging token-level uncertainty scores and aggregating them into claim-level scores. Moreover, we introduce a new token-level uncertainty score, namely claim-conditioned probability (CCP), which demonstrates confident improvements over several baselines for seven LLMs and four languages.

To the best of our knowledge, there is no previous work that has investigated the quality of claim-level UQ techniques for LLM generation. Therefore, for this purpose, we construct a novel benchmark based on fact-checking of biographies of individuals generated using a range of LLMs. Note that different LLMs produce different outputs, which generally have higher variability than, e.g., outputs in such tasks as machine translation or question answering. Therefore, we compare the predictions and uncertainty scores to the results of an automatic external fact-checking system FactScore (Min et al., 2023). Human evaluation verifies that our constructed benchmark based on FactScore can adequately evaluate the performance of the uncertainty scores.

Our contributions are as follows:

- We propose a *novel framework* for fact-checking LLM generations using token-level uncertainty quantification. We provide a procedure for efficiently estimating the uncertainty of atomic claims generated by a white-box model and highlighting potentially deceptive fragments by mapping them back to the original response.
- We propose a *novel method* for token-level uncertainty quantification that outperforms baselines and can be used as a plug-in in a fact-checking framework.
- We design a *novel approach to evaluation* of token-level UQ methods for white-box LLMs based on fact-checking, which can be applied to other white-box LLMs.
- We provide an *empirical and ablation analysis* of the method for fact-checking of LLM generations, and find that the uncertainty scores we produce can help to spot claims with factual errors for seven LLMs over four languages:

English, Chinese, Arabic, and Russian.

- The method is implemented as a part of the LM-Polygraph library (Fadeeva et al., 2023). All the code and data for experiments is publicly available¹.

2 Related Work

2.1 Fact-Checking LLM Generations and Detecting Hallucinations

The problem of hallucinations has made fact-checking of LLM outputs a prominent topic in the research community and resulted in a surge of publications on the topic. Chern et al. (2023) present Factool – a task and domain agnostic framework for hallucination detection that leverages GPT for claim extraction and verification. Manakul et al. (2023) suggest to sample multiple outputs from black-box LLMs and evaluate how similar the sampled responses are using the external model. Varshney et al. (2023) detect LLM hallucinations by extracting key parts of output using an external model and estimating their uncertainty based on logits. The most uncertain parts are verified using an external knowledge source. Pan et al. (2023) propose to fact-check complex statements by decomposing them into simpler subtasks and generating reasoning programs to verify these statements. Min et al. (2023) present a methodology for evaluating long LLM-generated texts by decomposing them into simple atomic statements and further verifying them against some knowledge source. Several subsequent works further optimize components of the knowledge-based fact-checking pipelines (Wang et al., 2024a,b).

In contrast to previous work that leverages external knowledge sources for fact-checking (a database or another LLM), our work is the first to investigate token-level UQ methods for this task using the LLM and its outputs only.

2.2 Uncertainty Quantification of LLM Generations

UQ techniques for LLM generation can be classified into five major categories (Fadeeva et al., 2023). Information-based methods leverage the probability distribution of generated tokens and usually do not require any additional models. In this category, we can include methods such as perplexity (Fomicheva et al., 2020), mean token entropy (Fomicheva et al., 2020), point-wise mutual

¹<https://github.com/IINemo/lm-polygraph>

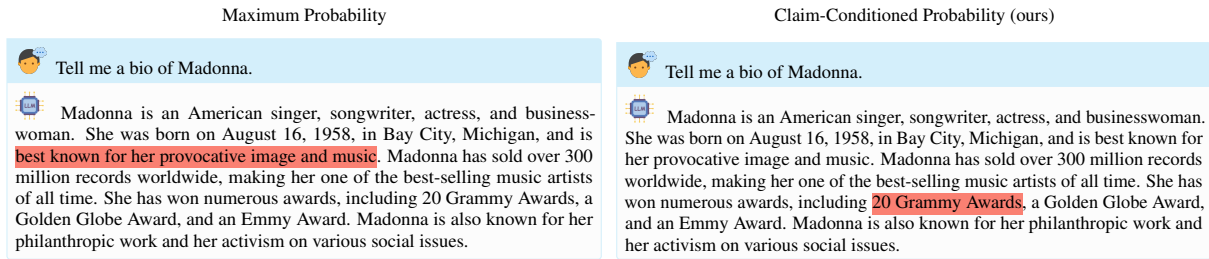


Figure 1: Visual comparison of our Claim-Conditioned Probability method to the Maximum Probability baseline. CCP accurately identifies the incorrectly specified number of awards (in red), whereas Maximum Probability erroneously highlights the claim that is actually correct.

information (PMI) (Takayama and Arase, 2019), and conditional PMI (van der Poel et al., 2022).

Another category of methods is based on density estimation of latent instance representations. A typical example in this category is Mahalanobis distance (Lee et al., 2018) and its various modifications (Ren et al., 2023; Vazhentsev et al., 2023b). The disadvantage of such methods is the need for access to the LLM training data in order to fit external density models, which is problematic for most general-purpose LLMs.

Ensembling and Monte Carlo dropout methods are based on the lexical diversity of multiple outputs sampled from one or multiple versions of LLMs for a single query (Malinin and Gales, 2020; Fomicheva et al., 2020). Their main drawback is that they require many predictions, which makes them too computationally and memory intensive for practical purposes. Additionally, it is difficult to apply these techniques to quantify uncertainty of text fragments such as claims, as different samples may seriously diverge.

It has recently been shown that LLMs can reflexively estimate the confidence of their generations simply by asking themselves about the truthfulness of their output (Kadavath et al., 2022). This can work better than analyzing the probability distribution for the original prediction, but requires a second pass of inference, feeding the original output as a part of the query.

Finally, there is a group of methods that leverages the diversity of meanings that the LLM generates for a given query. This group includes semantic entropy (Kuhn et al., 2023) and various scores based on the analysis of the similarity matrix between outputs (Lin et al., 2023).

UQ methods can also be classified into white-box and black-box (Lin et al., 2023) approaches,

depending on the required access to LLM itself and its outputs. Black-box techniques do not require any other input except generated texts.

The method in our work can be attributed to the information-based group, and can be applied only to white-box LLMs because it requires access to the probability distribution of the generated tokens. Compared to other techniques, it offers a novel approach to post-processing the probability distribution and is specifically designed to quantify uncertainty of output fragments, such as atomic claims and individual words.

3 Fact-Checking Pipeline

The fact-checking pipeline (see Figure 4) starts with splitting a generated text into atomic claims, e.g. using a much smaller model fine-tuned for this particular task. For experimental evaluations in this work, we follow the FactScore approach (Min et al., 2023), where splitting is implemented via the OpenAI Chat API.

Each atomic claim is matched against the sequence of tokens in the original text with the corresponding probability distributions. Then we calculate token-level uncertainty scores and aggregate them into the claim-level uncertainty.

Finally, the claim-level uncertainty scores are compared against a threshold obtained on a validation set to determine whether the claim should be highlighted for the end-user as unreliable. Individual tokens can be a part of multiple atomic claims. If the token belongs to a reliable and unreliable claim at the same time, it is not highlighted. An example visualization is presented in Figure 1.

4 Uncertainty Quantification

In this section, we first provide background on common UQ methods that can be used at the token level,

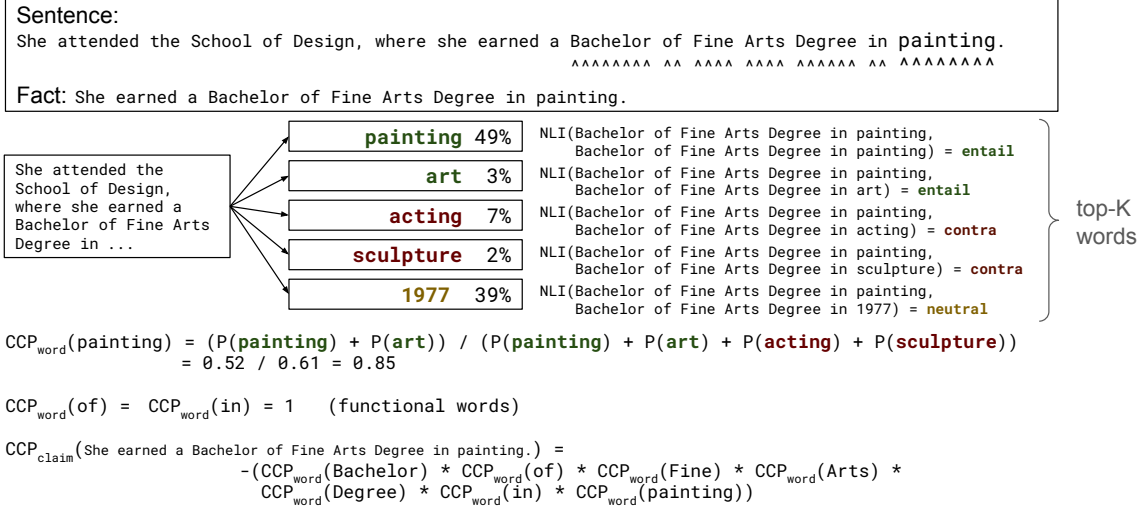


Figure 2: Example of CCP calculation for the word *painting* in a Vicuna 13b generation.

then delve into our **Claim-Conditioned Probability (CCP)** token-level method, and finally describe how token-level uncertainties are aggregated into a claim-level score.

Autoregressive language models generate text token by token. In this work, we will operate on the level of words and without loss of generality suppose that the autoregressive distribution at each step generates the random word $X_j \sim P(\cdot | x_{<j})$, where $x_{<j}$ is the text generated before the word at position j . We also denote by x_j the generated word at position j and by $x_{1:j} = x_{<j} \circ x_j$ a text composed of words at positions 1 to j . For example, in the case of greedy generation, $x_j = \arg \max_x P(x | x_{<j})$, where x_j is the most probable realization of X_j . Let us also denote by C a set of indices of words corresponding to a particular atomic claim.

4.1 Claim-Level UQ Baselines

We note that for UQ to be practical, it needs to be fast. Therefore, we do not consider methods such as deep ensembles (Lakshminarayanan et al., 2017), due to their significant computational overhead.

Maximum Probability represents a basic approach to UQ, where we simply treat the probability of the most likely generation as a confidence score:

$$MP(C) = 1 - \prod_{j \in C} P(x_j | x_{<j}). \quad (1)$$

Perplexity is a common metric used to evaluate the performance of LLMs. Lower perplexity indicates that a model’s probability distribution better predicts a sample. It is computed as the average

negative log probability of generated tokens that belong to the claim C :

$$Perp(C) = \exp \left(-\frac{1}{|C|} \sum_{j \in C} \log P(x_j | x_{<j}) \right). \quad (2)$$

Maximum Entropy of a token in the claim:

$$Ent(C) = \max_{j \in C} \mathcal{H}(\cdot | x_{<j}), \quad (3)$$

where $\mathcal{H}(\cdot | x_{<j})$ is the entropy of the autoregressive distribution of the current token. Preliminary experiments indicated that simply getting the maximum of the token entropies in a claim noticeably outperforms other aggregation techniques like average or minimum. It is also generally a slightly better baseline than perplexity.

P(True), similar to (Kadavath et al., 2022), measures the uncertainty of the claim by asking the LLM itself whether the generated claim is true or not. The confidence is the probability of the first generated token y_1 being equal to “True”:

$$P(\text{True}) = 1 - P(y_1 = \text{“True”}). \quad (4)$$

While some work has reported that this technique outperforms other baselines, the big drawback is that one needs to run the original LLM twice.

4.2 Claim-Conditioned Probability

In this subsection, we propose a novel method for token- and claim-level uncertainty quantification.

4.2.1 Motivation and Theoretical Background

When an LLM generates an output, it faces various types of uncertainty reflected in the token distribution of the current generation step (see Figure 5 for

an example). We identify three distinct types of uncertainty:

(1) Claim type/order uncertainty: What claim to generate on the current step? For example, on the current step, an LLM might hesitate between generating a year of graduation of a person and a field of study. A different order of claims, missing claims, or different types of generated claims do not make produced text less factual. Therefore, when performing fact-checking, we should not take this type of uncertainty into account.

(2) Surface form uncertainty: What synonyms or hypernyms to use when generating a claim (e.g. “art” or “painting”)? Different surface forms also do not make the text less factual — they might change the style, but not the underlying meaning of the text. Therefore, this type of uncertainty is also not relevant for fact-checking.

(3) Claim uncertainty: What specific piece of information to relay for a particular claim type? For example, an LLM might not be sure which field of study to generate, producing a token distribution with multiple highly-probable variants, such as “painting”, “acting”, “sculpture”. Similarly, for the year of graduation, an LLM might produce a distribution with various potential years. This uncertainty is relevant for fact-checking, because if the model is not sure about the information it relays, there might be a high chance of a factual mistake.

Two out of the three types of uncertainty are irrelevant for fact-checking and only introduce noise into the final score. We propose a new UQ method that ignores the first two types of uncertainty and focuses only on the third one, namely **Claim-Conditioned Probability (CCP)**:

$$CCP(x_j) = P(\text{Meaning}(x_{1:j}) | x_{<j}, \text{ClaimType}(x_{1:j})). \quad (5)$$

Here, $\text{ClaimType}(x_{1:j})$ represents a claim type of the generated sequence $x_{1:j}$ and $\text{Meaning}(x_{1:j})$ is a function that maps x_j into its meaning given the previous words in a sentence $x_{<j}$, so that various surface forms with a similar meaning for x_j are mapped to a single categorical variable.

Conditional probability can be rewritten using unconditional probabilities:

$$\begin{aligned} & P(\text{Meaning}(x_{1:j}) | \text{ClaimType}(x_{1:j}), x_{<j}) \\ &= \frac{P(\text{Meaning}(x_{1:j}), \text{ClaimType}(x_{1:j}) | x_{<j})}{P(\text{ClaimType}(x_{1:j}) | x_{<j})}. \end{aligned}$$

Assuming that each meaning in a word distribution can correspond to only a single claim type, the joint

probability is the same as the meaning probability $P(\text{Meaning}(x_{1:j}), \text{ClaimType}(x_{1:j}) | x_{<j}) = P(\text{Meaning}(x_{1:j}) | x_{<j})$.

Meaning probability in turn sums from the probabilities of word alternatives x_j^k that correspond to the same meaning: $P(\text{Meaning}(x_{1:j}) | x_{<j}) = \sum_{x_j^k \in M(x_j)} P(x_j^k | x_{<j})$, where we say that $x_j^k \in M(x_j)$ if $\text{Meaning}(x_{1:j}) = \text{Meaning}(x_{<j} \circ x_j^k)$.

In the same way, the probability of a claim type sums from probabilities of all meanings and transitionally from probabilities of words that correspond to the particular claim type: $P(\text{ClaimType}(x_{1:j})) = \sum_{x_j^l \in CT(x_j)} P(x_j^l | x_{<j})$, where we denote by $x_j^l \in CT(x_j)$ an event such that $\text{ClaimType}(x_{1:j}) = \text{ClaimType}(x_{<j} \circ x_j^l)$. Therefore, equation (5) can be rewritten as follows:

$$CCP(x_j) = \frac{\sum_{x_j^k \in M(x_j)} P(x_j^k | x_{<j})}{\sum_{x_j^l \in CT(x_j)} P(x_j^l | x_{<j})}. \quad (6)$$

The meaning function and the construction of the set of words that belong to the same claim type can be implemented in various ways. We outline our approach in Section 4.2.2.

Previously-proposed UQ methods have partially accounted for some of the types of uncertainty described above. For example, semantic entropy (Kuhn et al., 2023) accounts for uncertainty in semantically-equivalent groups, which helps to alleviate the surface-form uncertainty. However, in our method, we additionally remove the impact of claim-type uncertainty.

4.2.2 Implementation

We implement CCP using NLI at the word level. We compare the original claim and the claim, in which the target word is replaced by its alternatives from the autoregressive distribution.

The distribution X_j at the position j is approximated by top- K alternatives $\{x_j^k\}_{k=1}^K$ with $x_j^1 \equiv x_j$. We replace x_j with its alternatives x_j^k and obtain new instances $x_{<j} \circ x_j^k, k = 1, \dots, K$. Each new instance is compared against the original prediction $x_{1:j} = x_{<j} \circ x_j$ using an NLI model. We define $\text{NLI}(x_j^k, x_j) := \text{NLI}(x_{<j} \circ x_j^k, x_{1:j})$, where $\text{NLI}(x_{<j} \circ x_j^k, x_{1:j})$ means application of the NLI model to the text fragments $x_{<j} \circ x_j^k$ and $x_{1:j}$.

The outcome of the NLI procedure is one of three labels: entail (‘e’), contradict (‘c’), or neutral (‘n’). If the new instance entails the origi-

nal prediction $\text{NLI}(x_j^k, x_j) = \text{'e'}$, then we consider $x_{<j} \circ x_j^k$ has the same meaning with $x_{1:j}$ ($x_j^k \in M(x_j)$) and corresponds to the same claim type ($x_j^k \in CT(x_j)$).

If the new instance contradicts the original prediction $\text{NLI}(x_j^k, x_j) = \text{'c'}$, then we consider $x_{<j} \circ x_j^k$ has a different meaning with $x_{1:j}$ ($x_j^k \notin M(x_j)$), but corresponds to the same claim type ($x_j^k \in CT(x_j)$). Otherwise, if the new instance is neutral w.r.t. the original prediction $\text{NLI}(x_j^k, x_j) = \text{'n'}$, then we consider that $x_{<j} \circ x_j^k$ does not correspond to the same claim type as $x_{1:j}$ ($x_j^k \notin CT(x_j)$). Thus, equation (6) for CCP can be written as follows:

$$CCP_{word}(x_j) = \frac{\sum_{k:\text{NLI}(x_j^k, x_j)=\text{'e'}} P(x_j^k | x_{<j})}{\sum_{k:\text{NLI}(x_j^k, x_j) \in \{\text{'e'}, \text{'c'}\}} P(x_j^k | x_{<j})}.$$

For practical considerations, we consider that CCP for function words is always equal to 1. In our experiments, we base this determination on the stop word list from NLTK (Bird and Loper, 2004).

We note that most transformer LLMs generate sub-word tokens instead of whole words. To obtain distributions for whole words, we generate one or multiple tokens using beam search with K beams.

To obtain CCP-based claim-level uncertainty, we simply take the product of CCPs of each word from the claim C :

$$CCP_{claim}(C) = 1 - \prod_{j \in C} CCP_{word}(x_j). \quad (7)$$

An example of calculating the CCP for a claim is presented in Figure 2. Other detailed examples of CCP calculation are available in Appendix B.

5 Benchmark for Evaluation of Claim-Level UQ Methods

We evaluate claim-level UQ techniques and their ability to spot hallucinations on the task of generating biographies. In relevant previous work (Manakul et al., 2023), the authors generate biographies with one LLM (GPT-3), manually annotate sentences for factuality, and quantify uncertainty of a different “proxy” model. Factuality labels are then used to evaluate the quality of uncertainty scores. We argue that such an approach based on a proxy model introduces a big discrepancy between the generated text and what a proxy LLM actually wants to generate, which results in biased UQ evaluation results. To make the evaluation as close

as possible to the real-world scenario, we allow unrestricted generation of biographies from LLMs.

Unrestricted generation complicates automatic evaluation of the fact-checking pipeline, because obtaining gold standard annotation requires manual annotation of all outputs from each model. Therefore, in addition to manual annotation, we annotate claims in generated texts automatically using FactScore – a fact-checking tool (Min et al., 2023), which has access to an external knowledge source. Using FactScore, enables completely automatic evaluation and allows us to scale up experiments.

We generate LLM responses in English, Chinese, Arabic, and Russian to 100 biography prompts. The typical biography prompt is *Give me a biography for <person name>* in different languages. The set of people was generated by asking GPT-4 to list the most famous people since 1900. The maximum generation length is set to 256 tokens. If the last sentence of the generation is unfinished (i.e. does not end with any punctuation), it is discarded. We generate responses for the following LLMs: (for English) Vicuna 13b (Zheng et al., 2023), Mistral 7b (Jiang et al., 2023), Jais 13b (Sengupta et al., 2023), and GPT-3.5-turbo (Ouyang et al., 2022); (for Chinese) Yi 6b (Young et al., 2024); (for Arabic) Jais 13b and GPT-4; (for Russian) Vikhr-instruct-0.2 7b (Nikolich et al., 2024).

We decompose the generated text into atomic claims using GPT-4. For each claim, we map all its words back to generated text to access the corresponding token logits. Not all claims perfectly match to the original response. For example, for Vicuna 13b around 5% of all claims do not successfully match because ChatGPT abstained to respond or outputted words not present in the original. We consider only successfully matched claims.

For English, we are able to perform annotation completely automatically: each atomic claim is classified by FactScore as supported or not supported. The underlying FactScore model is “retrieval+ChatGPT” with a dump of Wikipedia articles as an external knowledge source. We also manually annotated 100 claims from English biographies produced by the Vicuna 13b model, 183 claims from Arabic biographies, 146 claims from Russian biographies, and 1603 claims in Chinese. Each of the statements was checked by two annotators with access to the corresponding Wikipedia article. The final label is set to “supported” only if both annotators label it as so.

The statistics of the resulting datasets with

Model	Mistral 7b	Vicuna 13b	Jais 13b	GPT-3.5-turbo
CCP (ours)	0.66 \pm 0.03	0.66 \pm 0.04	0.71 \pm 0.05	0.58 \pm 0.04
Maximum Prob.	0.59 \pm 0.03	0.60 \pm 0.05	0.64 \pm 0.05	0.54 \pm 0.05
Perplexity	0.58 \pm 0.07	0.58 \pm 0.07	0.61 \pm 0.08	0.53 \pm 0.06
Token Entropy	0.60 \pm 0.02	0.60 \pm 0.06	0.63 \pm 0.06	0.53 \pm 0.03
P(True)	0.53 \pm 0.02	0.61 \pm 0.03	0.55 \pm 0.05	0.53 \pm 0.04

Table 1: ROC-AUC of claim-level UQ methods with FactScore labels as the ground truth (English).

Model	Mistral 7b	Vicuna 13b	Jais 13b	GPT-3.5-turbo
CCP (ours)	0.34 \pm 0.05	0.24 \pm 0.04	0.33 \pm 0.07	0.14 \pm 0.01
Maximum Prob.	0.26 \pm 0.04	0.18 \pm 0.05	0.24 \pm 0.05	0.13 \pm 0.02
Perplexity	0.27 \pm 0.03	0.17 \pm 0.04	0.21 \pm 0.05	0.11 \pm 0.03
Token Entropy	0.30 \pm 0.06	0.18 \pm 0.04	0.24 \pm 0.06	0.12 \pm 0.02
P(True)	0.24 \pm 0.03	0.21 \pm 0.06	0.19 \pm 0.06	0.17 \pm 0.02

Table 2: PR-AUC (considering the Not Supported class as positive) of claim-level UQ methods with FactScore labels as the ground truth (English).

automatically-labeled claims in English are presented in Table 6, and the statistics of manually annotated datasets are presented in Table 7. The majority of claims in the model output are correct, with 6–29% of hallucinations. The automatic pipeline for evaluation of UQ methods using FactScore is illustrated in Figure 6 in Appendix C.

6 Experiments

6.1 Experimental Setup

Fact-checking of atomic claims is framed as a binary classification task, where uncertainty scores serve as predictors of non-factuality, and FactScore or human labels serve as ground truth. The evaluation metric is ROC-AUC and PR-AUC (unsupported claims as a positive class).

For the CCP method, NLI scores are calculated using the DeBERTa-large model (He et al., 2021) fine-tuned for this task². The number of alternatives used in CCP is $K = 10$, except for GPT-3.5-turbo and GPT-4, as the OpenAI API does not allow us to retrieve more than 5 alternatives from the token distribution. Details on hardware and computational resources spent for experiments can be found in Appendix F.

6.2 Results for English on the FactScore Annotation

The main results of the experiments with FactScore labels for English are presented in Tables 1 and 2.

²<https://huggingface.co/microsoft/deberta-large-mnli>

The proposed CCP method outperforms all other UQ techniques for each of the considered LLMs, with only exception in the PR-AUC metric for the GPT-3.5-turbo model, where the P(True) approach exhibits the highest performance. The underperformance of CCP for GPT-3.5-turbo may be attributed to the limited number of token options and their associated logits available through the OpenAI API. The best overall improvement from CPP is obtained for Jais 13b, where it outperforms the closest competitor by 0.07 ROC-AUC and 0.09 PR-AUC.

We further analyze the performance by plotting ROC-AUC as a function of the number of considered sentences from the beginning of the generated text (see Figure 3). We note that the quality of each method decreases as the number of considered sentences increases. This may be due to the fact that the model tends to start the response with easy-to-know and hence reliable claims and as it generates more text, it has to produce more complex and less reliable statements, which is illustrated for the Vicuna 13b model in Figure 8 in the appendix. For the majority of cases, CCP outperforms other methods, except when we consider only the first two and the first five sentences generated by GPT-3.5-turbo.

6.3 Multilingual Results on Manual Annotation

Multilingual results based on manual annotation are presented in Tables 3 and 4 and in Figure 9. Using manual annotation for English, we can also evaluate the performance of FactScore itself. The accuracy of automatic annotation is 77.2%

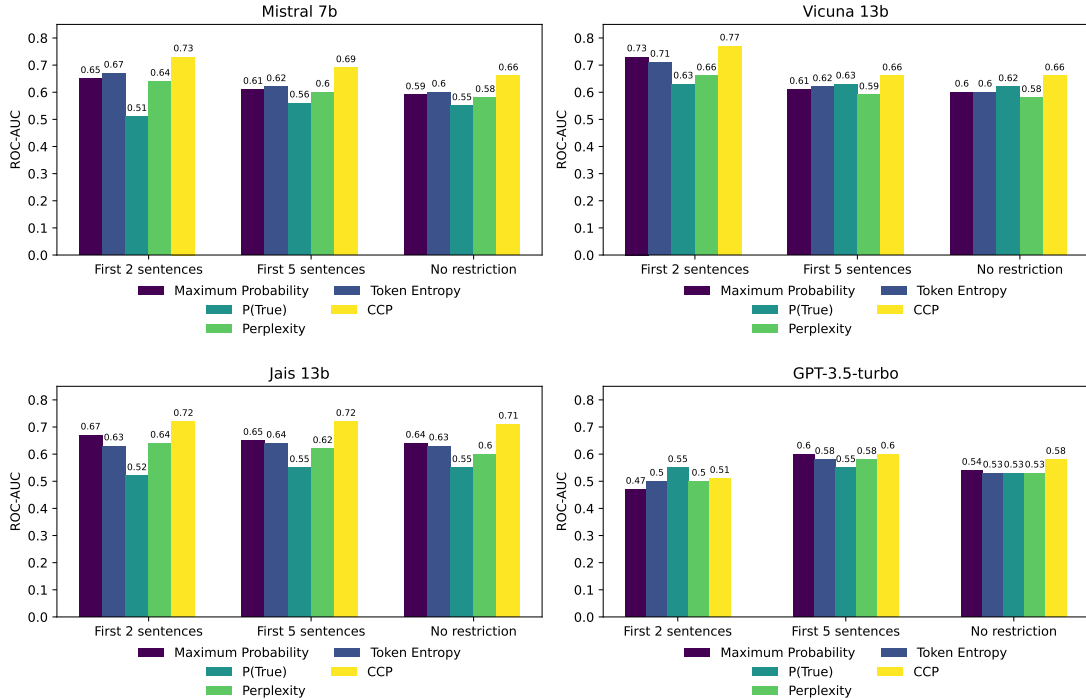


Figure 3: ROC-AUC of claim-level UQ methods based on FactScore labels, aggregated into bins when considering only facts from the first 2, 5, and all sentences (English).

and ROC-AUC is 0.72. A detailed analysis of FactScore mistakes is presented in Appendix C.3.

For human annotation, the performance of all UQ methods appears to be even slightly higher than for the labels obtained using FactScore (Table 3). Moreover, CCP outperforms FactScore itself by 0.06 ROC-AUC. These results demonstrate that in the task of detecting LLM hallucinations, UQ techniques can be a strong alternative to fact-checking tools with an external knowledge source.

For Chinese, Arabic, and Russian, CCP also outperforms the baselines. For the Chinese Yi 6b model, from Figure 9, we can see that the gap between CCP and the baselines is especially significant for the several first claims. When considering more claims, CCP still clearly outperforms the Maximum Probability baseline, but the P(True) baseline substantially reduces the gap. For Arabic and Jais, CCP outperforms the closest competitor by 0.05 ROC-AUC. On GPT-4 generations for Arabic, the metrics for all methods are low. We explain this observation by a small ratio of non-factual claims in the GPT-4 output. For Russian generations with the Vikhr model, CCP confidently outperforms Maximum Probability, which is its closest competitor, by 0.05 ROC-AUC.

	Ground-Truth	
Method	Human	FactScore
CCP (ours)	0.78	0.74
Maximum Prob.	0.67	0.65
Perplexity	0.65	0.64
Token Entropy	0.69	0.65
P(True)	0.68	0.65
FactScore	0.72	–

Table 3: ROC-AUC of claim-level UQ methods with human annotation and FactScore annotation as the ground truth (English, Vicuna 13b model).

Model	Yi 6b, Chinese	Jais 13b, Arabic	GPT-4, Arabic	Vikhr 7b, Russian
CCP (ours)	0.64 ± 0.03	0.66 ± 0.02	0.56 ± 0.05	0.68 ± 0.04
Maximum Prob.	0.52 ± 0.03	0.59 ± 0.02	0.55 ± 0.08	0.63 ± 0.04
Perplexity	0.51 ± 0.04	0.56 ± 0.02	0.54 ± 0.08	0.58 ± 0.04
Token Entropy	0.57 ± 0.05	0.61 ± 0.02	0.48 ± 0.06	0.55 ± 0.03
P(True)	0.63 ± 0.04	0.61 ± 0.02	0.50 ± 0.06	0.58 ± 0.03

Table 4: ROC-AUC of claim-level UQ methods with manual annotation as the ground truth.

6.4 Ablation Studies

In this section, we analyze the influence of various CPP components on English biographies annotated with FactScore (Tables 9–12). Details of the experimental setup for each ablation study are presented in Appendix D.

(1) **Aggregation** of CCP_{word} for obtaining CCP_{claim} . Besides the product of probabilities, we also tried the normalized product, minimum, and

average probability. All these approaches perform slightly worse than the product (see Table 9).

(2) NLI model. We investigate the influence of the specific NLI model on the performance of CCP. Table 10 shows that CCP’s effectiveness is not critically dependent on the complexity of the NLI model employed. Notably, even a relatively small model with 22M parameters maintains strong performance without any degradation.

(3) NLI context. We analyze what context is sufficient for NLI in CCP (Table 11). In addition to the standard variant in CCP, where we keep the claim that precedes the word in question, we experiment with a single target word without context and the whole sentence that precedes the target word. All variants demonstrate lower performance. No context results in a drop of 0.02 ROC-AUC, and longer contexts – of more than 0.07.

(4) Functional words handling. The results in Figure 12 show that excluding functional words in CCP helps to improve the performance by 0.03 ROC-AUC. This approach also slightly improves the maximum probability baseline, but its performance is still much lower.

(5) The number of alternatives K . Taking $K = 5$ alternatives instead of $K = 10$ in CCP decreases ROC-AUC by 0.02. Figure 7 also demonstrates that further decreasing K reduces the performance even more. When increasing K , the performance plateaus at $K = 8$.

6.5 Qualitative Analysis

In qualitative analysis of uncertainty scores for various generations and models, we note that the maximal probability baseline produces a lot more false positives than CCP. This happens because CCP ignores some types of uncertainty, focusing only on the claim uncertainty. In some cases, CCP also finds false claims overlooked by other methods because ignoring certain types of uncertainty also allows us to reduce the cut-off threshold used to mark claims. An example where we compare CCP with maximal probability is presented in Figure 12, and more examples can be found in Appendix G.

6.6 Computational Efficiency

To demonstrate the computational efficiency of CCP, we compare it to the fastest UQ method – Maximum Probability. Experiments were conducted using a dataset of 100 biographies and Mistral 7b (Jiang et al., 2023). To ensure a fair comparison, we focus solely on the runtime of gen-

Method	NLI model parameters	Runtime
<i>MP</i>	—	18.5 ± 0.8 sec
<i>CCP</i>	350M	20.1 ± 0.9 sec
<i>CCP</i>	22M	19.1 ± 0.8 sec

Table 5: The runtime of Maximum Probability and CCP methods on 100 biographies in English.

erating biographies and calculating the respective uncertainty scores for each claim. Time spent on claim extraction and matching is excluded. The experiments utilized two 32GB V100 GPUs. Each biography was processed in a single batch.

MP does not introduce notable overhead over the generation process, as it only aggregates produced logits. CCP involves running an NLI model for each of 10 token candidates per token position, which introduces some overhead.

Table 5 presents the runtime comparison. We see that NLI does not make a substantial impact. Using the default microsoft/deberta-large-nli model (350M parameters) results in 8% increase of the runtime compared to MP. Using the smaller cross-encoder/nli-deberta-v3-xsmall model (22M parameters), which achieves comparable UQ performance, reduces the computational overhead to only 3%.

7 Conclusion

We presented a novel approach to fact-checking and hallucination detection based on token-level uncertainty quantification. According to human evaluation, our approach is competitive with FactScore, a fact-checking tool that leverages an external knowledge source: we achieve similar or better results with access to only LLM outputs.

We proposed a computationally efficient token-level and claim-level UQ method, Claim Conditioned Probability that outperforms a number of baselines in fact-checking. In this method, we post-process the word distribution to mitigate the impact of uncertainty related to the variability of surface forms and uncertainty about what claim type to generate on the current step. In the constructed benchmark, where we detect hallucinations in biographies, CCP outperforms other methods for seven LLMs, including GPT-3.5-turbo and GPT-4, and four languages. We also demonstrate that computational overhead of CCP might be as low as 3% of the LLM inference runtime.

Limitations

While this work has been conducted according to experimental and methodological best practice, there are several potential limitations.

First, at the core of the approach lies a text entailment classifier. As it was originally pre-trained for a slightly different use-case, more careful analysis of its performance on diverse domains and genres should be carried out.

Second, the current implementation of the method makes use of OpenAI's GPT models for text segmentation and extraction of atomic facts, similarly to FactScore, which may not be practical in real applications. Replacing these components with cheaper open models should be feasible in principle, but is left for future work.

Third, part of our experimental results rely on a human evaluation, which may be subjective. We tried to mitigate this by creating detailed instructions for the annotators, but a larger-scale study with larger overlap would further strengthen the results.

Fourth, our uncertainty quantification is based on tokens, where taking into account also larger units such as noun or verb phrases as basic units of analysis may be better motivated linguistically.

Fifth, in this work, we do not consider the calibration of CCP scores. We note that calibration on its own does not provide the information about the performance in the practical task that we are interested in – fact-checking. At the same time, CCP could be post-calibrated in the same way as any other probabilities or their surrogates.

Finally, our approach only detects potentially spurious generations. A prominent direction for future research is to modify the generation of an LLM to exclude such spans, while ensuring fluency of a generated text: a simple removal of uncertain claims may result in incoherent sentences.

Ethical Considerations

We would like to caution that our method for fact-checking is based on uncertainty quantification, and thus it is not bullet-proof, as it only reflects the internal model state of the LLM. Thus, it is of limited utility when it comes to claims that are beyond the time cutoff of the model. Moreover, the LLM might be trained on factually false data, which is beyond our control. It could be also tricked by variations in the prompt.

We also caution that our solution does not eliminate hallucinations; instead, it can be used to highlight risky parts of a text, for humans to take into account. Our intended use is towards raising awareness and promoting human-machine collaboration.

Finally, our method can be misused to unfairly moderate content. Thus, we ask researchers and potential users to exercise due caution.

References

- Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. [A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity](#). In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718, Nusa Dua, Bali. Association for Computational Linguistics.
- Steven Bird and Edward Loper. 2004. [NLTK: The natural language toolkit](#). In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.
- I-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, Pengfei Liu, et al. 2023. [FacTool: Factuality detection in generative AI—a tool augmented framework for multi-task and multi-domain scenarios](#). *arXiv preprint arXiv:2307.13528*.
- David Dale, Elena Voita, Loic Barrault, and Marta R. Costa-jussà. 2023. [Detecting and mitigating hallucinations in machine translation: Model internal workings alone do well, sentence similarity Even better](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–50, Toronto, Canada. Association for Computational Linguistics.
- Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. 2023. [LM-Polygraph: Uncertainty estimation for language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 446–461, Singapore. Association for Computational Linguistics.
- Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Spezia. 2020. [Unsupervised quality estimation for neural machine translation](#). *Transactions of the Association for Computational Linguistics*, 8:539–555.

- Yarin Gal et al. 2016. *Uncertainty in deep learning*. Ph.D. thesis, University of Cambridge.
- Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. *A survey on automated fact-checking*. *Transactions of the Association for Computational Linguistics*, 10:178–206.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. *DeBERTa: decoding-enhanced BERT with disentangled attention*. In *9th International Conference on Learning Representations*.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. *Mistral 7b*. *arXiv preprint arXiv:2310.06825*.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, et al. 2022. *Language models (mostly) know what they know*. *arXiv preprint arXiv:2207.05221*.
- Adam Tauman Kalai and Santosh S Vempala. 2023. *Calibrated language models must hallucinate*. *arXiv preprint arXiv:2311.14648*.
- Nikita Kotelevskii, Aleksandr Artemenkov, Kirill Fedyanin, Fedor Noskov, Alexander Fishkov, Artem Shelmanov, Artem Vazhentsev, Aleksandr Petiushko, and Maxim Panov. 2022. *Nonparametric uncertainty quantification for single deterministic neural network*. *Advances in Neural Information Processing Systems*, 35:36308–36323.
- Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. *Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation*. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. *Simple and scalable predictive uncertainty estimation using deep ensembles*. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NeurIPS 2017*, page 6405–6416, Red Hook, NY, USA. Curran Associates Inc.
- Moritz Laurer, Wouter van Atteveldt, Andreu Salleras Casas, and Kasper Welbers. 2022. *Less Annotating, More Classifying – Addressing the Data Scarcity Issue of Supervised Machine Learning with Deep Transfer Learning and BERT - NLI*. Preprint. Publisher: Open Science Framework.
- Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. *A simple unified framework for detecting out-of-distribution samples and adversarial attacks*. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. *Generating with confidence: Uncertainty quantification for black-box large language models*. *arXiv preprint arXiv:2305.19187*.
- Andrey Malinin and Mark Gales. 2020. *Uncertainty estimation in autoregressive structured prediction*. In *International Conference on Learning Representations*.
- Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. *SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models*. *arXiv preprint arXiv:2303.08896*.
- Sewon Min, Kalpesh Krishna, Xinxin Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. *FActScore: Fine-grained atomic evaluation of factual precision in long form text generation*. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100.
- Preslav Nakov, Giovanni Da San Martino, Tamer Elsayed, Alberto Barr  n-Cedeno, Rub  n M  guez, Shaden Shaar, Firoj Alam, Fatima Haouari, Maram Hasanain, Nikolay Babulkov, et al. 2021. *The clef-2021 checkthat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news*. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II 43*, pages 639–649. Springer.
- Aleksandr Nikolich, Konstantin Korolev, and Artem Shelmanov. 2024. *Vikhr: The family of open-source instruction-tuned large language models for russian*. *arXiv preprint arXiv:2405.13929*.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. *Training language models to follow instructions with human feedback*. *Advances in Neural Information Processing Systems*, 35:27730–27744.
- Liangming Pan, Xiaobao Wu, Xinyuan Lu, Anh Tuan Luu, William Yang Wang, Min-Yen Kan, and Preslav Nakov. 2023. *Fact-checking complex claims with program-guided reasoning*. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6981–7004.
- Nils Reimers and Iryna Gurevych. 2019. *Sentence-BERT: Sentence embeddings using siamese BERT-networks*. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J Liu. 2023. *Out-of-distribution detection and selective generation for conditional language models*. In *The Eleventh International Conference on Learning Representations*.

- Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Xudong Han, Soudos Mahmoud Bsharat, Alham Fikri Aji, Zhiqiang Shen, Zhengzhong Liu, Natalia Vassilieva, Joel Hestness, Andy Hock, Andrew Feldman, Jonathan Lee, Andrew Jackson, Hector Xuguang Ren, Preslav Nakov, Timothy Baldwin, and Eric Xing. 2023. [Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models](#). *arXiv preprint arXiv:2308.16149*.
- Jiao Sun, Yufei Tian, Wangchunshu Zhou, Nan Xu, Qian Hu, Rahul Gupta, John Wieting, Nanyun Peng, and Xuezhe Ma. 2023a. [Evaluating large language models on controlled generation tasks](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3155–3168. Association for Computational Linguistics.
- Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023b. [Is chatgpt good at search? Investigating large language models as re-ranking agent](#). *arXiv preprint arXiv:2304.09542*.
- Junya Takayama and Yuki Arase. 2019. [Relevant and informative response generation using pointwise mutual information](#). In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 133–138. Association for Computational Linguistics.
- Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023. [Large language models in medicine](#). *Nature medicine*, 29(8):1930–1940.
- Liam van der Poel, Ryan Cotterell, and Clara Meister. 2022. [Mutual information alleviates hallucinations in abstractive summarization](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5956–5965. Association for Computational Linguistics.
- Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jian-shu Chen, and Dong Yu. 2023. [A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation](#). *arXiv preprint arXiv:2307.03987*.
- Artem Vazhentsev, Gleb Kuzmin, Artem Shelmanov, Akim Tsvigun, Evgenii Tsymbalov, Kirill Fedyanin, Maxim Panov, Alexander Panchenko, Gleb Gusev, Mikhail Burtsev, Manvel Avetisian, and Leonid Zhukov. 2022. [Uncertainty estimation of transformer predictions for misclassification detection](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8237–8252, Dublin, Ireland. Association for Computational Linguistics.
- Artem Vazhentsev, Gleb Kuzmin, Akim Tsvigun, Alexander Panchenko, Maxim Panov, Mikhail Burtsev, and Artem Shelmanov. 2023a. [Hybrid uncertainty quantification for selective text classification in ambiguous tasks](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11659–11681, Toronto, Canada. Association for Computational Linguistics.
- Artem Vazhentsev, Akim Tsvigun, Roman Vashurin, Sergey Petrakov, Daniil Vasilev, Maxim Panov, Alexander Panchenko, and Artem Shelmanov. 2023b. [Efficient out-of-domain detection for sequence to sequence models](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1430–1454, Toronto, Canada. Association for Computational Linguistics.
- David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. [Fact or fiction: Verifying scientific claims](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550.
- Hao Wang, Yuxia Wang, Minghan Wang, Yilin Geng, Zhen Zhao, Zenan Zhai, Preslav Nakov, Timothy Baldwin, Xudong Han, and Haonan Li. 2024a. [Loki: An open-source tool for fact verification](#).
- Yuxia Wang, Revanth Gangi Reddy, Zain Muhammad Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, Isabelle Augenstein, Iryna Gurevych, and Preslav Nakov. 2024b. [Factcheck-bench: Fine-grained evaluation benchmark for automatic fact-checkers](#). *arXiv preprint arXiv:2311.09000*.
- 01.AI: Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. [Yi: Open foundation models by 01.ai](#).
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. [Judging LLM-as-a-judge with MT-Bench and Chatbot Arena](#). *arXiv preprint arXiv:2306.05685*.

A Fact-Checking Pipeline

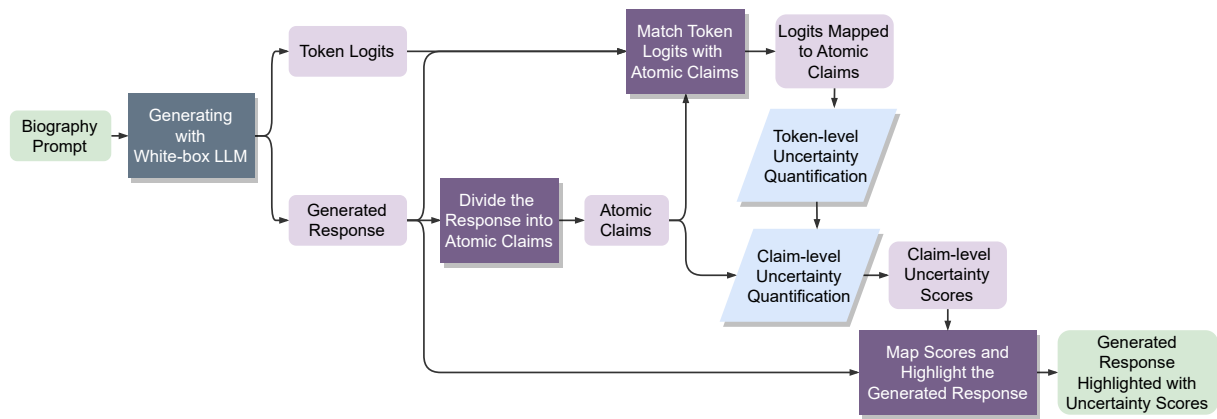


Figure 4: The scheme of the fact-checking pipeline based on UQ.

B Example of CCP Calculation

Figure 5 demonstrates an example of the LLM generation process and CCP calculation process. CCP quantifies the Claim Uncertainty, not taking into account the Claim order and Surface form uncertainties. As a result, CCP produces better uncertainty scores than Maximum Probability.

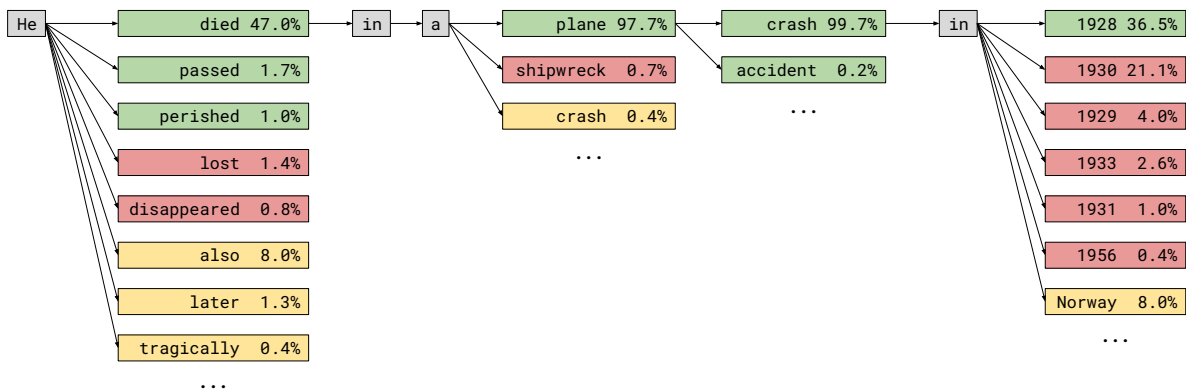


Figure 5: Example of the Vicuna 13b generation process and CCP calculation process part. The words from the greedy-generated sentence are presented sequentially on the top, each non-functional word is supplemented with its alternatives and autoregressive generation probabilities. Words with probability less than 0.1% are omitted. Green-colored words indicate entailment to the greedy generated word, red color indicates contradiction, and yellow color indicates neutral NLI class. On the last position, CCP successfully distinguishes Norway from other year-related words, and does not consider its probability in the final formula.

C Benchmark Construction Details

C.1 Benchmark Construction Pipeline

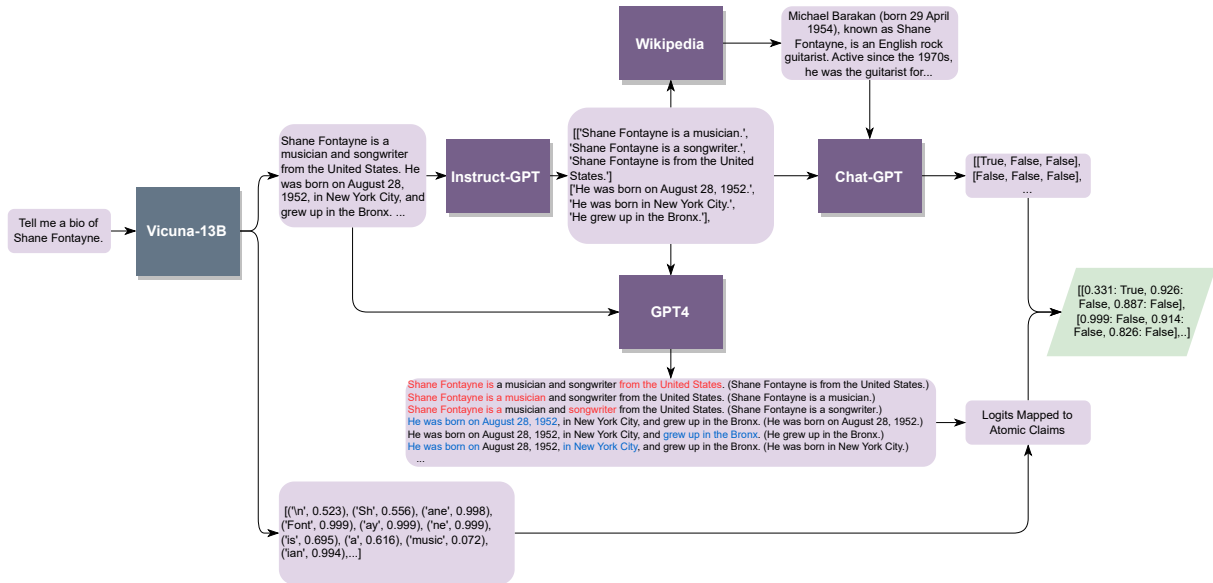


Figure 6: A visualization of the benchmark construction pipeline based on FactScore with an example.

Figure 4 presents the suggested general preparation pipeline of a factuality dataset for an arbitrary model and Fact-Checking benchmark. This pipeline was used to generate the biography dataset. Figure 6 presents the detailed pipeline of the biography dataset preparation, based on the general schema.

The prompt for the matching model to map atomic claims to the initial text is as follows: “Given the fact “fact”, identify the corresponding words in the original sentence “sent” that help derive this fact. Please list all words that are related to the fact, in the order they appear in the original sentence, each word separated by comma.”. The prompt for the model that partitions the initial generated text into individual atomic claims, as well as the prompt for the classification model, are taken similar to the paper by [Min et al. \(2023\)](#).

C.2 Datasets and Statistics

Model	Number of claims	Supported claims
Mistral 7b	3,824	71.1%
Vicuna 13b	3,617	78.2%
Jais 13b	1,407	84.3%
GPT-3.5-turbo	3,875	89.4%

Table 6: The statistics of the datasets generated from 100 biographies using all tested English LLMs and annotated automatically.

Model	Number of claims	Supported claims
Vicuna 13b, English	100	70.1%
Yi 6b, Chinese	1,603	94.0%
Jais 13b, Arabic	186	73.0%
GPT-4, Arabic	200	92.5%
Vikhr 7b, Russian	146	72.6%

Table 7: The statistics of the datasets generated from 100 biographies using all tested English LLMs and annotated manually.

Since FactScore only supports English, for Arabic, Chinese, and Russian, we generate biographies of

well-known people and annotate them only manually. We also manually annotate 100 English claims generated by Vicuna 13b. The statistics for the annotated datasets are presented in Table 7.

For Arabic, using GPT-4, we generate 100 biographies of people randomly selected from the list of the most visited websites in Arabic Wikipedia. The used Arabic prompt is the translation of: “Tell me the biography of {person name}”. To extract claims, we prompt GPT-4 in the following way: “Convert the following biography into Arabic atomic factual claims that can be verified, one claim per line. Biography is: {biography}”. Arabic biographies and claims are translated into English using Google Translate. It is worth mentioning that almost one-third of the names in the list of person names are foreign, e.g. Donald Trump, Messi, Isaac Newton, Pope Benedict XVI, etc. On average, GPT-4 generates 20 claims from each biography, and random two claims from each biography are verified manually (total = 200 claims).

For Jais 13b experiments, we use the same prompts used for GPT-4. We notice that the biographies generated by Jais 13b are much shorter than the ones generated by GPT-4 (almost half-length). Similarly, we use GPT-4 to extract claims from the generated biographies. On average, biographies generated by Jais 13b have nine claims. Jais 13b generates empty biographies for seven names (out of 100) with response messages like: “I am sorry! I cannot provide information about {name}”, or “What do you want to know exactly?”. Two random claims from each biography are verified manually (total = 186 claims).

For Chinese, we first prompt ChatGPT to generate a list of 100 famous people. Then use the same way as we have done in Arabic, but change the prompt to Chinese, to generate biographies and claims. We use Yi 6b to generate texts and GPT-4 to split them into atomic claims.

For Russian, we conduct a similar approach, prompting ChatGPT to generate a list of 100 famous people and checking the result to obtain representative personalities from different areas such as science, sport, literature, art, government activity, cinematography, heroes, etc. A balanced list of famous people in different professional categories was obtained. For these people, we generate biographies using the Vikhr 7b model (Nikolich et al., 2024).

C.3 FactScore Annotation

The statistics of English data annotated using FactScore is presented in Table 6. Here we give examples of the operation of the FactScore automatic markup system, see the Table 8. We hypothesize that the causes of FactScore errors are related to model hallucination (true information is present in the knowledge source, but the model produces incorrect information), lack of context (an excerpt from a Wikipedia article cannot capture all the details), and difficulties with information interpretation (the desired information is present in the knowledge base, but is formulated in different words or in several sentences). All these cases leave room for further improvement of the pipeline. We have also given the results of the proposed CCP method on selected examples. It can be seen that the results of our algorithm are similar to the annotation from FactScore, while CCP does not use external information in calculating the scores.

D Ablation Studies

In this section, we perform the ablation study of various CCP components.

D.1 Aggregation

Table 9 presents the result of CCP on Vicuna 13b when applying 4 different kinds of aggregation:

$$\begin{aligned}
 CCP_{prod}(F) &= 1 - \prod_{j=1}^k CCP_{word}(x_j), \\
 CCP_{len}(F) &= 1 - \exp\left(\frac{1}{k} \sum_{j=1}^k \log CCP_{word}(x_j)\right), \\
 CCP_{min}(F) &= 1 - \min_{j=1, \dots, k} CCP_{word}(x_j), \\
 CCP_{mean}(F) &= 1 - \frac{1}{k} \sum_{j=1}^k CCP_{word}(x_j).
 \end{aligned}$$

Cases	FS	Human	CCP	Generated Atomic Claim	True Information from the Wikipedia article
TN	False	False	0.819	Marie Stopes died on October 20, 1958.	Marie Charlotte Carmichael Stopes (15 October 1880 – 2 October 1958) was a British author...
TN	False	False	0.999	Planck is best known for his work on the nature of light.	His fame as a physicist rests primarily on his role as the originator of quantum theory...
FN	False	True	1.0	Heisenberg was appointed as the director of the Max Planck Institute.	He then became director of the Max Planck Institute for Physics and Astrophysics from 1960 to 1970.
FN	False	True	0.716	Ray Charles incorporated elements of Latin music into his sound.	Charles reached the pinnacle of his success at Atlantic with the release of "What'd I Say", which combined gospel, jazz, blues and Latin music.
FP	True	False	0.001	Sagan was a prolific writer.	Carl Edward Sagan (November 9, 1934 – December 20, 1996) was an American astronomer and science communicator.
FP	True	False	0.141	Van Gogh was a pastor.	Van Gogh prepared for the University of Amsterdam theology entrance examination; he failed the exam ...
FP	True	False	0.185	Hawking showed an early aptitude for science.	Although known at school as "Einstein", Hawking was not initially successful academically.
TP	True	True	0.276	Tiger Woods is a professional golfer.	Eldrick Tont "Tiger" Woods (born December 30, 1975) is an American professional golfer.
TP	True	True	0.015	Miles Davis began playing the trumpet at the age of 13.	On his thirteenth birthday his father bought him a new trumpet,[17] and Davis began to play in local bands.
TP	True	True	0.001	Hitchcock directed "The Birds."	Hitchcock's other notable films include Rope (1948), Strangers on a Train (1951), ..., Birds (1963) and Marnie (1964), ...

Table 8: Table with examples of the FactScore automatic annotation system and all types of classification outcomes when comparing automatic annotation and manual annotation (confusion matrix elements): True Negative (TN), False Negative (FN), False Positive (FP), True Negative (TN); FS is a FactScore label, Human is a human annotation label. CCP scores are comparable to FactScore annotation and do not require an external source of information.

Method	ROC-AUC	PR-AUC
CCP_{prod}	0.66 \pm 0.03	0.22 \pm 0.05
CCP_{len}	0.65 \pm 0.03	0.21 \pm 0.03
CCP_{min}	0.64 \pm 0.03	0.13 \pm 0.04
CCP_{mean}	0.65 \pm 0.03	0.22 \pm 0.04

Table 9: ROC-AUC and PR-AUC on Vicuna 13b generation, for different normalizations.

CCP_{prod} shows the best results and is chosen for the final CCP formula.

D.2 NLI Models

Table 10 presents the result of CCP on Vicuna 13b when using different NLI models. We selected fine-tuned NLI models from HuggingFace, encompassing a range of sizes from Microsoft’s DeBERTa model (He et al., 2021), a multilingual variant (Laurer et al., 2022), and a model from SentenceTransformers (Reimers and Gurevych, 2019).

Our findings demonstrate that the CCP performance exhibits minimal dependence on the specific NLI model chosen. Notably, even CCP utilizing a relatively small CrossEncoder model with only 22M parameters achieves the highest performance.

D.3 NLI Context

Table 11 presents the result of CCP on Vicuna 13b when adding different contexts as the inputs of NLI model:

1. $CCP_{no\ context}$ calculates NLI using these 2 words as NLI model input;

NLI model	Parameters	ROC-AUC	PR-AUC
microsoft/deberta-large-mnli	350M	0.66 \pm 0.06	0.24 \pm 0.04
microsoft/deberta-base-mnli	86M	0.65 \pm 0.06	0.23 \pm 0.05
MoritzLaurer/mDeBERTa-v3-base	86M	0.66 \pm 0.06	0.21 \pm 0.03
cross-encoder/nli-deberta-v3-large	350M	0.66 \pm 0.06	0.22 \pm 0.04
cross-encoder/nli-deberta-v3-base	86M	0.65 \pm 0.06	0.23 \pm 0.05
cross-encoder/nli-deberta-v3-small	44M	0.65 \pm 0.06	0.22 \pm 0.05
cross-encoder/nli-deberta-v3-xsmall	22M	0.66 \pm 0.07	0.24 \pm 0.06

Table 10: The ROC-AUC and PR-AUC metrics of CCP on biographies generation dataset with Vicuna 13b model, when using different NLI models with specified identifiers in Huggingface library.

Method	ROC-AUC	PR-AUC
$CCP_{no\ context}$	0.64 ± 0.03	0.24 ± 0.04
$CCP_{sent\ pref}$	0.59 ± 0.03	0.18 ± 0.04
$CCP_{claim\ pref}$	0.66 ± 0.03	0.24 ± 0.05

Table 11: ROC-AUC and PR-AUC on Vicuna 13b generation, for different contexts to input words with to an NLI model.

- $CCP_{sent\ pref}$ calculates NLI using the prefix in the sentence from model generation up to the current word and its alternative, as input to NLI model;
- $CCP_{claim\ pref}$, uses all words in the sentence corresponding to the greedy word, which were matched to the current fact and which comes before the greedy word (see example on Figure 2).

The results for sentence prefix provides too wide context to the NLI model, which results in the NLI model focusing on the contexts instead of the greedy words and its alternatives, thus outputting lots of entailment classes. On the other hand, the method without context does not provide enough context to calculate NLI class with more quality. $CCP_{claim\ pref}$ shows better results and is the one chosen for the final CCP formula.

D.4 Functional Words Handling

Method	ROC-AUC	PR-AUC
$CCP_{confident}$	0.66 ± 0.03	0.24 ± 0.05
CCP_{ignore}	0.63 ± 0.03	0.23 ± 0.06
$MP_{confident}$	0.59 ± 0.05	0.18 ± 0.04
MP_{ignore}	0.60 ± 0.05	0.19 ± 0.05

Table 12: ROC-AUC and PR-AUC on Vicuna 13b generation, for methods of handling functional words.

Table 12 presents the result of CCP and Maximum Probability on Vicuna 13b when handling functional words differently. The 2 approaches tested are:

- $CCP_{confident}$ and $MP_{confident}$ are the versions of CCP and MP baselines, which assigns the most confident word-level score of 1.0 to words from NLTK (Bird and Loper, 2004) stopwords list, which is the same as skipping these words from the list of matched words of length k ;
- CCP_{ignore} and MP_{ignore} handles functional words similar to any other words. MP_{ignore} is the baseline used in the main section.

$CCP_{confident}$ shows better results and is the one chosen for final CCP formula. In contract, Maximum Probability performs worse if specifically handling functional words.

D.5 Number of Alternatives

Figure 7 presents the CCP ROC-AUC results on the whole model generation, depending on the number of beams n to run the method with.

E Additional Experimental Results

Here we show additional results related to the performance of our pipeline and CCP method. In the Figure 8 we show the dependence of the percentage of supported claims in the generated LLM response (in this case for Vicuna 13b) as a function of its length. As we can see, the LLM produces wrong claims when the generation length increases, which may be due to the generation of additional facts.

In Figure 9, we show the performance quality of our CCP method on Chinese data on the Yi 6b model. We see that, as for other English models, our method outperforms the alternatives over the entire generation length in terms of ROC-AUC.

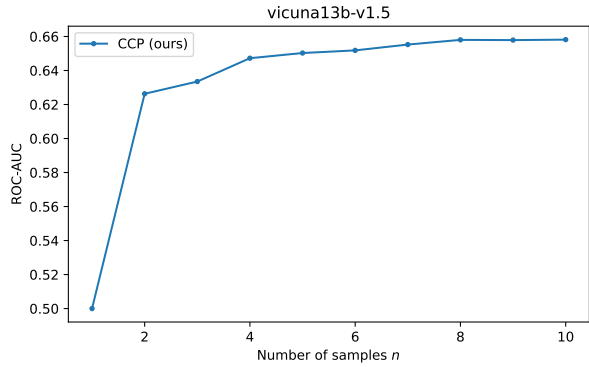


Figure 7: ROC-AUC between FactScore classes and the claim-level CCP method as a function of n – the number of token options in the probability distribution of the LLM to consider.

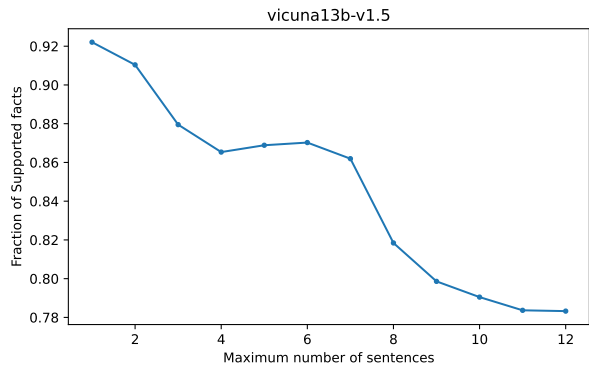


Figure 8: Percentage of supported claims, as a function of the number of sentences to restrict generation to (English, Vicuna 13b).

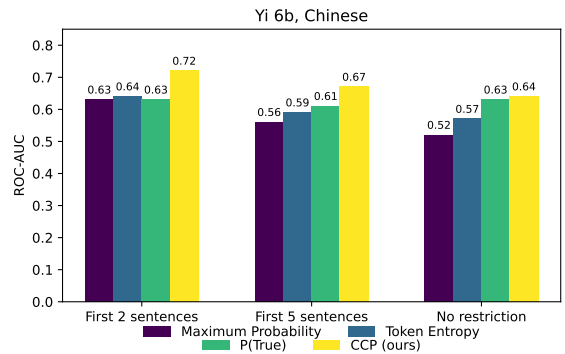


Figure 9: The comparison of token-level uncertainty quantification methods in terms of ROC-AUC scores, measured for Chinese dataset. The results are split into bins when considering only facts from the first 2, 5, and all sentences.

F Resources and Expenses

For a single run of data generation and UQ methods evaluation, we spent 12 days of Nvidia A100 GPU compute time. OpenAI API was used mainly for splitting and matching atomic claims, the total cost for all English models (Mistral 7b, Vicuna 13b, Jais 13b, GPT-3.5-turbo) was \$40. The cost of generating English biographies with GPT-3.5-turbo was \$13.

G Examples of Results Obtained with CCP and the Maximal Probability Baseline

Figures 10, 11 and 12 show examples of the Claim-Conditioned Probability method compared with Maximum Probability. On all examples, our CCP method correctly detects the wrong facts, whereas Maximum Probability mistakenly highlights correct facts.



Figure 10: The Maximum Probability baseline vs Claim-Conditioned Probability (ours).

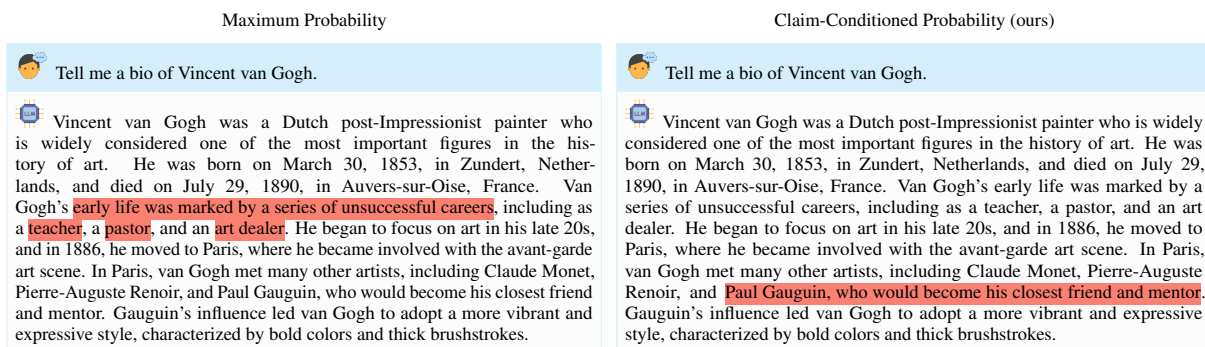


Figure 11: The Maximum Probability baseline vs Claim-Conditioned Probability (ours).

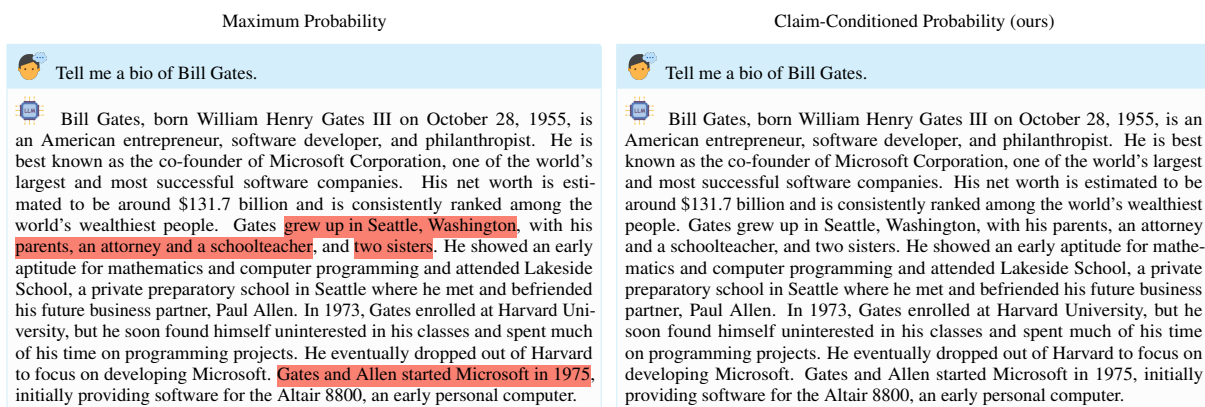


Figure 12: The Maximum Probability baseline vs Claim-Conditioned Probability (ours).