STAT 689: Large-scale and high-dimensional statistical inference **Fall 2024**

Lecture 15

In this lecture, we will discuss applications of some of the techniques we have learned in the context of large language models. In particular, we shall focus on the following two topics:

- Detecting AI-Generated Content Using Watermarking;
- Test Sets Contamination;
- Guaranteeing Factual Reliability of Outputs.

1 Detecting AI-Generated Content Using Watermarking

With the increasing use of large language models in recent years, it has become essential to differentiate between text generated by these models and text written by humans. Some of the most advanced LLMs, such as GPT-4, Llama 3, and Gemini, are very good at producing human-like texts, which could be challenging to distinguish from human-generated texts, even for humans. However, it is crucial to distinguish between human-produced texts and machine-produced texts to prevent the spread of misleading information, improper use of LLM-based tools in education, model extraction attacks through distillation, and the contamination of training datasets for future language models.

1.1 Generative watermarking

Watermarking is a principled method for embedding nearly unnoticeable statistical signals into text generated by LLMs, enabling provable detection of LLM-generated content from its human-written counterpart. This work focuses on a scenario where an untrusted third-party user sends prompts to a trusted large language model (LLM) provider, who then generates a text from their LLM with a watermark. This makes it possible for a detector to later identify the source of the text if the user publishes it. The user is allowed to modify the generated text by making substitutions, insertions, or deletions before publishing it.

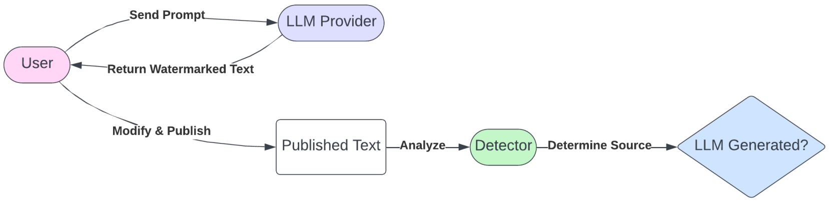


Figure 1: An illustration of generative watermarking for LLMs.

1.2 Watermarked text generation

Denote by $\mathcal{V}$ the vocabulary, and let $\mathcal{P}$ be an autoregressive LLM which maps a string $y_{-n_0:t-1} = y_{-n_0}y_{-n_0+1}\cdots y_{t-1} \in \mathcal{V}^{t+n_0}$ to a distribution over the vocabulary, with $p(\cdot|y_{-n_0:t-1})$ being the distribution of the next token y_t. Here $y_{-n_0:0}$ denotes the prompt provided by the user.

For ease of notation, we will assume that $\mathcal{V} = \{1, 2, \ldots, V\}$, where V is the vocabulary size. Let $\xi_{1:t} = \xi_1\xi_2\cdots\xi_t$ be a watermark key sequence with $\xi_i \in \Xi$ for each i, where Ξ is a general space. Given a prompt sent from a third-party user, the LLM provider calls a generator to autoregressitvely generate text from an LLM using a decoder function Γ, which maps ξ_t and a distribution p_t over the next token to a value in $\mathcal{V}$.

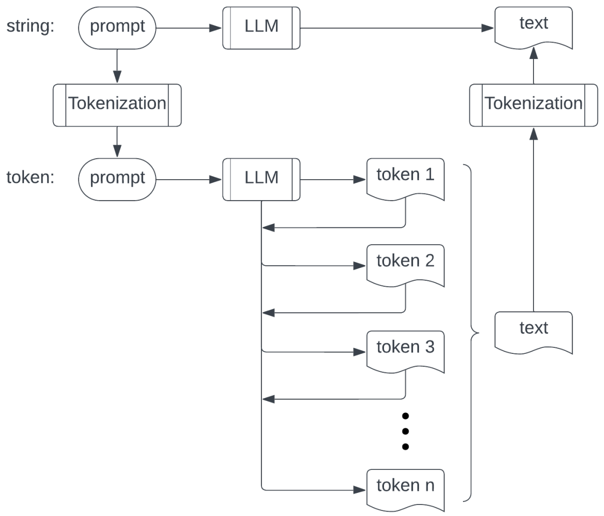


Figure 2: A simplified illustration of autoregressive LLMs.

The watermarking scheme should preserve the original text distribution, i.e., $P(\Gamma(\xi_t, p_t) = y) = p_t(y)$. A watermark text generation algorithm recursively generates a string $y_{1:n}$ by

$$y_i = \Gamma(\xi_i, p(\cdot|y_{-n_0:i-1})), \quad 1 \leq i \leq n,$$

where n is the number of tokens in the text $y_{1:n}$ generated by the LLM, and ξ_i's are assumed to be independently generated from some distribution ν over Ξ. In other words, given $p(\cdot|y_{-n_0:i-1})$, y_i is completely determined by ξ_i and $y_{-n_0:i-1}$.

Definition. A watermarking scheme is distortion-free if

$$P(y_i = k) = p(k|y_{-n_0:i-1}).$$

In other words, the watermarking scheme preserves the original text distribution.

Example 1. Consider a simple setting where $\mathcal{V} = \{0, 1\}$. A simple way to generate y_i is by setting

$$y_i = \mathbf{1}\{\xi_i \leq p(1|y_{-n_0:i-1})\},$$

where ξ_i is generated from the uniform distribution. It is clear that

$$P(y_i = 1) = P(\xi_i \leq p(1|y_{-n_0:i-1})) = p(1|y_{-n_0:i-1}).$$

Therefore, this watermarking scheme is distortion-free.

Example 2. We discuss another scheme called exponential minimum sampling (EMS) proposed in Aaronson (2023). To generate each token of a text, we first sample $\xi_{ik} \sim \text{Unif}[0, 1]$ independently for $1 \leq k \leq V$. Let

$$y_i = \arg\max_{1\leq k\leq V} \frac{\log(\xi_{ik})}{p(k|y_{-n_0:i-1})} = \arg\min_{1\leq k\leq V} \frac{-\log(\xi_{ik})}{p(k|y_{-n_0:i-1})} = \arg\min_{1\leq k\leq V} E_{ik},$$

where $E_{ik} := -\log(\xi_{ik})/p(k|y_{-n_0:i-1}) \sim \text{Exp}(p(k|y_{-n_0:i-1}))$ with $\text{Exp}(a)$ denoting an exponential random variable with the rate a.

Exercise 15.1. For two exponential random variables $X \sim \text{Exp}(a)$ and $Y \sim \text{Exp}(b)$, we have two basic properties: (i) $\min(X, Y) \sim \text{Exp}(a + b)$; (ii) $P(X < Y) = \mathbb{E}[1 - \exp(-aY)] = a/(a + b)$. Using (i) and (ii), show that

$$P(y_i = k) = P\left(E_{ik} < \min_{j\neq k} E_{ij}\right) = p(k|y_{-n_0:i-1}).$$

Hence, EMS preserves the original text distribution.

1.3 Watermark detection

We now consider the detection problem, which involves determining whether a given text is watermarked or not. Consider the case where a string $\widetilde{y}_{1:m}$ is published by the third-party user, and a key sequence $\xi_{1:n}$ is provided to a detector. The detector calls a detection method to test

$$H_0 : \widetilde{y}_{1:m} \text{ is not watermarked} \quad \text{versus} \quad H_a : \widetilde{y}_{1:m} \text{ is watermarked},$$

by computing a p-value with respect to a test statistic $\phi(\xi_{1:n}, \widetilde{y}_{1:m})$. It is important to note that the text published by the user can be quite different from the text initially generated by the LLM using the key $\xi_{1:n}$, which we refer to as $y_{1:n}$. To account for this difference, we can use a transformation function $\mathcal{E}$ that takes $y_{1:n}$ as the input and produces the published text $\widetilde{y}_{1:m}$ as the output, i.e.,

$$\widetilde{y}_{1:m} = \mathcal{E}(y_{1:n}).$$

This transformation can involve substitutions, insertions, deletions, paraphrases, or other edits to the input text.

The test statistic ϕ measures the dependence between the text $\widetilde{y}_{1:m}$ and the key sequence $\xi_{1:n}$. Throughout our discussions, we will assume that a large value of ϕ provides evidence against the null hypothesis (e.g., stronger dependence between $\widetilde{y}_{1:m}$ and $\xi_{1:n}$). To obtain the p-value, we consider a randomization test. In particular, we generate $\xi_i^{(t)} \sim \nu$ independently over $1 \leq i \leq n$ and $1 \leq t \leq T$, and $\xi_i^{(t)}$s are independent with $\widetilde{y}_{1:m}$. Then the randomization-based p-value is given by

$$p_T = \frac{1}{T+1}\left(1 + \sum_{t=1}^{T} \mathbf{1}\{\phi(\xi_{1:n}, \widetilde{y}_{1:m}) \leq \phi(\xi_{1:n}^{(t)}, \widetilde{y}_{1:m})\}\right).$$

Exercise 15.2. Show that p_T is super uniform, i.e., $P(p_T \leq x) \leq x$ for any $x \in [0, 1]$.

As the published text can be modified, it is not expected that every token in $\widetilde{y}_{1:m}$ will be related to the key sequence. Instead, we expect certain sub-strings of $\widetilde{y}_{1:m}$ to be correlated with the key sequence under the alternative hypothesis H_a. To measure the dependence, we use a scanning method that looks at every segment/sub-string of $\widetilde{y}_{1:m}$ and a segment of $\xi_{1:n}$ with the same length B. We use a measure $\mathcal{M}(\xi_{a:a+B-1}, \widetilde{y}_{b:b+B-1})$ to quantify the dependence between $\xi_{a:a+B-1}$ and $\widetilde{y}_{b:b+B-1}$, chosen based on the watermarked text generation method described above. Examples of $\mathcal{M}$ include

- Pearson correlation;
- Rank correlation;
- Edit distance;
- Other distance used in computational linguistics.

Given $\mathcal{M}$ and the block size B, we can define the maximum test statistic as

$$\phi(\xi_{1:n}, \widetilde{y}_{1:m}) = \max_{1 \leq a \leq n-B+1} \max_{1 \leq b \leq m-B+1} \mathcal{M}(\xi_{a:a+B-1}, \widetilde{y}_{b:b+B-1}).$$

The maximum statistic is more sensitive to a watermarked sub-string (think about the maximum test or the Bonferroni correction).

2 Test Sets Contamination

Large language models (LLMs) have significantly improved performance on various natural language processing benchmarks. These advancements are primarily due to extensive pretraining using massive datasets gathered from the internet. However, because of the minimal curation of these datasets, concerns have arisen regarding dataset contamination—where the pretraining dataset includes certain evaluation benchmarks.

This contamination complicates our ability to accurately assess the true performance of language models, raising questions about whether they are simply memorizing answers to challenging exam questions.

Understanding the balance between generalization and memorization of test sets is crucial for evaluating language model performance. Unfortunately, this task has become increasingly challenging, as many of the pretraining datasets used in contemporary language models are rarely made public.

In this section, we will consider a recent statistical method that aims to identify the presence of a benchmark in the pre-training dataset of an LLM with provable Type I error rate guarantees without access to the training data. To ensure this, one approach is to leverage the property of exchangeability found in many datasets. This property means that the order of examples within the dataset can be shuffled without affecting its joint distribution. The insight is that if a language model shows a preference for a specific ordering of the dataset—such as a canonical ordering found in publicly available repositories—it indicates a violation of exchangeability. This preference can only arise if the model has observed the dataset during training.

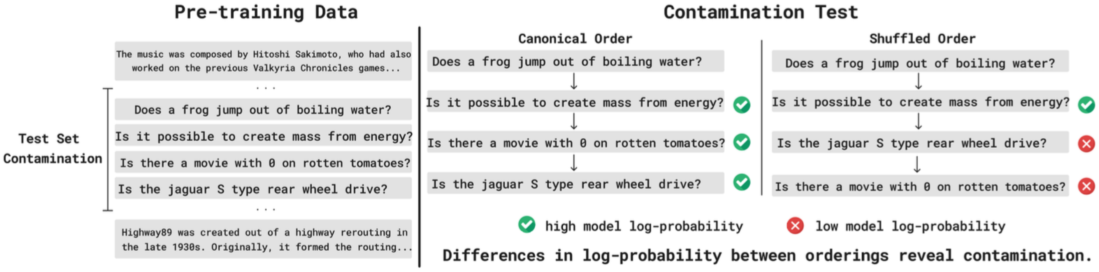

Pre-training Data
The music was composed by Hitoshi Sakimoto, who had also worked on the previous Valkyria Chronicles games...
...
Test Set Contamination
Does a frog jump out of boiling water?
Is it possible to create mass from energy?
Is there a movie with 0 on rotten tomatoes?
Is the jaguar S type rear wheel drive?
...
Highway89 was created out of a highway rerouting in the late 1930s. Originally, it formed the routing...
Contamination Test
Canonical Order
Does a frog jump out of boiling water?
Is it possible to create mass from energy? ✓
Is there a movie with 0 on rotten tomatoes? ✓
Is the jaguar S type rear wheel drive? ✓
Shuffled Order
Does a frog jump out of boiling water?
Is it possible to create mass from energy? ✓
Is the jaguar S type rear wheel drive? ✗
Is there a movie with 0 on rotten tomatoes? ✗
✓ high model log-probability
✗ low model log-probability
Differences in log-probability between orderings reveal contamination.

Figure 3: Test for contamination based on the canonical order.

Suppose we want to identify whether the training process of an LLM $\mathcal{P}$ included a dataset X. Statistically, this can be formulated as a hypothesis-testing problem:

$$H_0 : \mathcal{P} \text{ is independent of } X \quad \text{versus} \quad H_a : \mathcal{P} \text{ is dependent on } X,$$

where we treat $\mathcal{P}$ as a random variable whose randomness arises from a combination of the draw of the pretraining dataset (potentially including X). We write $X = (X_1, \ldots, X_n)$, where X_i represents the ith example in the dataset. Given a permutation π of $(1, 2, \ldots, n)$, we define

$$X_\pi = (X_{\pi(1)}, \ldots, X_{\pi(n)}).$$

We let $\mathcal{P}(X) = \mathcal{P}(X_1, \ldots, X_n)$ be the probability that the LLM generates the text X. We assume that the exchangeability of X, i.e.,

$$X \overset{d}{=} X_\pi$$

for any permutation π.

Lemma. For an exchangeable X and under the H_0, we have

$$\log \mathcal{P}(X) \overset{d}{=} \log \mathcal{P}(X_\pi).$$

Proof. Since X is exchangeable and $\mathcal{P}$ is independent of X, we must have

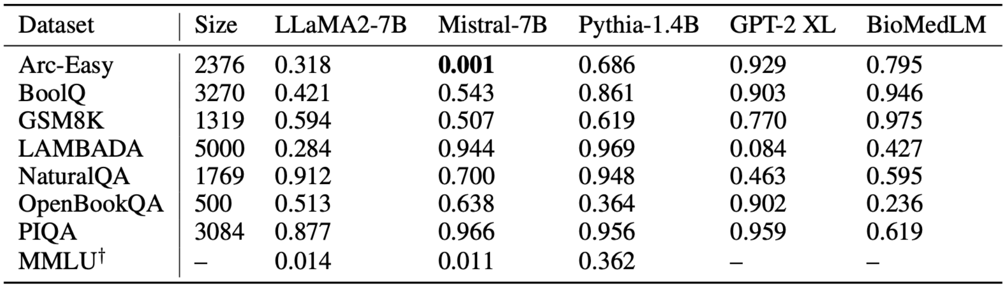
$$(\mathcal{P}, X) \overset{d}{=} (\mathcal{P}, X_\pi),$$

which implies that $\log \mathcal{P}(X) \overset{d}{=} \log \mathcal{P}(X_\pi)$.

To perform the test, we can calculate the p-value,

$$p = \frac{1 + \sum_{i=1}^{m} \mathbf{1}\{\log \mathcal{P}(X) \le \mathcal{P}(X_{\pi_i})\}}{m+1}.$$

based on m permutations $\pi_1, \ldots, \pi_m$. Under H_0, p is super uniformrm. Under the alternative, $\mathcal{P}(X)$ is expected to be larger than $\mathcal{P}(X_{\pi_i})$, and thus, the p-value will be small.



Dataset	Size	LLaMA2-7B	Mistral-7B	Pythia-1.4B	GPT-2 XL	BioMedLM
Arc-Easy	2376	0.318	**0.001**	0.686	0.929	0.795
BoolQ	3270	0.421	0.543	0.861	0.903	0.946
GSM8K	1319	0.594	0.507	0.619	0.770	0.975
LAMBADA	5000	0.284	0.944	0.969	0.084	0.427
NaturalQA	1769	0.912	0.700	0.948	0.463	0.595
OpenBookQA	500	0.513	0.638	0.364	0.902	0.236
PIQA	3084	0.877	0.966	0.956	0.959	0.619
MMLU†	–	0.014	0.011	0.362	–	–

Figure 4: P-values for contamination tests on open models and benchmarks.

3 Guaranteeing Factual Reliability of Outputs

LLMs have increasingly been adopted in various domains. However, LLM outputs cannot be fully trusted due to their tendency to generate hallucinations and non-factual content. In this section, we will learn how to use conformal prediction to enable such high-probability correctness guarantees for black-box LLMs.

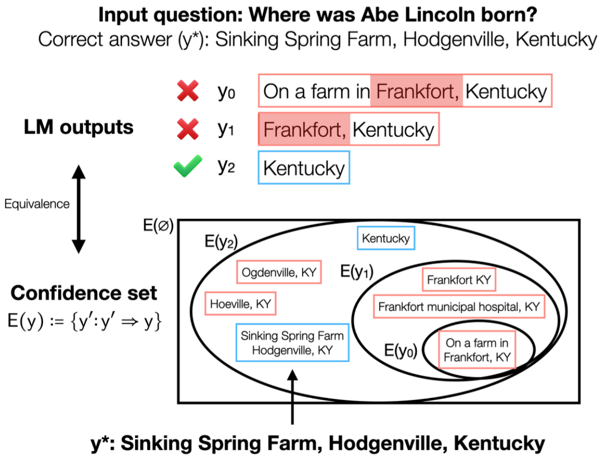

Input question: Where was Abe Lincoln born?
Correct answer (y*): Sinking Spring Farm, Hodgenville, Kentucky
LM outputs
y0
On a farm in Frankfort, Kentucky
y1
Frankfort, Kentucky
y2
Kentucky
Equivalence
E(∅)
E(y2)
Kentucky
Ogdenville, KY
E(y1)
Frankfort KY
Hoeville, KY
Frankfort municipal hospital, KY
Sinking Spring Farm Hodgenville, KY
E(y0)
On a farm in Frankfort, KY
Confidence set
E(y) := {y′:y′ ⇒ y}
y*: Sinking Spring Farm, Hodgenville, Kentucky

Figure 5: Conformal factuality uses conformal prediction to ensure the correctness of LLM outputs.

3.1 Correctness via entailments

In the standard text generation setting, an LLM $\mathcal{P}$ receives an input/prompt x from the user and generates an output $y = \mathcal{P}(x) \in \mathcal{Y}$, where $\mathcal{Y}$ is the space of outputs for the LLM. Similarly, we let $\mathcal{X}$ be the space of the inputs. Let $y^* \in \mathcal{Y}$ be a ground truth. A key problem with LLMs is that the output y may not be fully supported by y^*. As it is difficult to ensure the correctness of every LLM output, a more reasonable goal would be to provide high-probability guarantees such that for any user-specified probability $\alpha \in (0, 1)$, the LLM is correct with probability at least $1 - \alpha$ over some distribution $\mathbb{P}$. We express this goal as

$$\mathbb{P}(y \text{ is correct and factual}) \geq 1 - \alpha.$$

The first question is how we can formalize this correctness constraint. Here, we shall adopt the notion of entailments with respect to some reference knowledge y^* where the correctness is equivalent to the entailment relation

$$y^* \implies y.$$

Representing factuality and correctness via entailments to a reference is quite general, as we can set y^* to be a broad knowledge base such as "Wikipedia pages related to the input x" or even "all facts accessible via Google" to handle the case where there is no ground truth response for the input x.

The factuality constraint $y^* \implies y$ can be equivalently written as a set containment relation

$$y^* \in E(y) := \{y' \in \mathcal{Y} : y' \implies y\}.$$

We now show how the conformal inference could be useful in the current setting. In conformal prediction, we are given a set of exchangeable samples $(X_i, Y_i)_{i=1}^n$ and a future covariate X_{n+1}. The goal is to construct a prediction set $C(X_{n+1})$ such that

$$\mathbb{P}(Y_{n+1} \in C(X_{n+1})) \geq 1 - \alpha.$$

To utilize conformal inference, we shall replace $C(X_{n+1})$ with the entailment set $E(\mathcal{P}(X_{n+1}))$ of an LLM output $\mathcal{P}(X_{n+1})$ with the input X_{n+1}. Let Y_{n+1} be the ground truth associated with X_{n+1}. Our goal is thus to ensure that

$$\mathbb{P}(Y^*_{n+1} \in E(\mathcal{P}(X_{n+1}))) \geq 1 - \alpha,$$

where the event $Y^*_{n+1} \in E(\mathcal{P}(X_{n+1}))$ is equivalent to the event that $\mathcal{P}(X_{n+1})$ is correct according to Y^*_{n+1}.

3.2 Split conformal inference revisited

We revisit the split conformal inference approach and adapt it to the current setting. The description below is somewhat more general than what we have learned. Let $\{G_t(x)\}_{t\in\mathcal{T}}$ denote a sequence of output sets satisfying that $G_t(x) \subset G_{t'}(x)$ for $t \leq t'$. Consider the score

$$S(x, y) = \inf\{t \in \mathcal{T} : y \in G_t(x)\}. \tag{1}$$

This can be thought of as the minimum safe threshold where $y \in G_t(x)$ for every $t > S(x, y)$. Split conformal prediction then sets the final confidence set as

$$C(x) = G_{q_\alpha}(x),$$

where q_α is the $\lceil (n+1)(1-\alpha)\rceil/n$th quantile of these scores $S(X_i, Y_i)$ for $i = 1, 2, \ldots, n$. We note that

$$Y_{n+1} \in C(X_{n+1}) \Leftrightarrow Y_{n+1} \in G_{q_\alpha}(X_{n+1})$$

which then implies that

$$S(X_{n+1}, Y_{n+1}) \leq \lceil (1-\alpha)(n_2+1)\rceil \text{ smallest of } S(X_i, Y_i), i = 1, 2, \ldots, n.$$

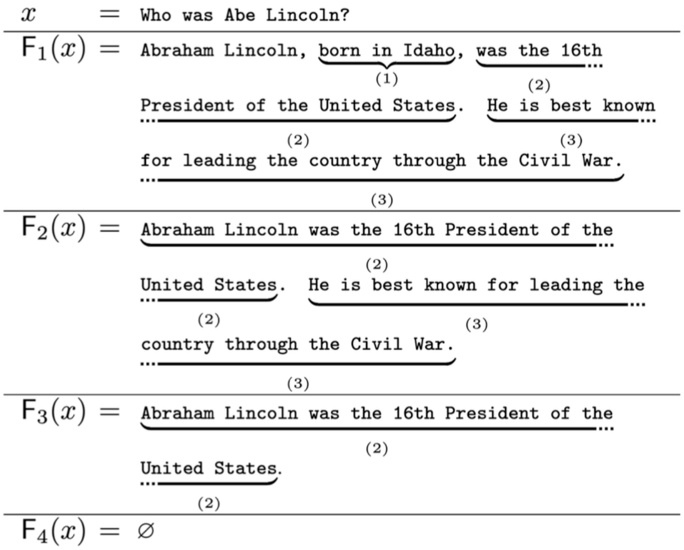
Using what we learn from Lecture 11, we have

$$\mathbb{P}(Y_{n+1} \in C(X_{n+1})) \geq 1 - \alpha.$$

3.3 Application to LLM

The correctness of an LLM output y is equivalent to the event $y^* \in E(y)$, and we seek to find some y that makes this event hold with probability at least $1 - \alpha$. To this end, we construct sequences of outputs $\{y_t\}_{t\in\mathcal{T}}$, which induces sequences of sets $\{E(y_t)\}_{t\in\mathcal{T}}$ on which we can apply conformal prediction.

To employ conformal prediction methods, we shall define the conformal sets $\{G_t(x)\}_{t\in\mathcal{T}}$ and the score $S(x, y)$. For the conformal sets, we will define these sets using the entailment operator E as $G_t(x) = E(F_t(x, \mathcal{P}(x)))$ where $F_t : \mathcal{X} \times \mathcal{Y} \mapsto \mathcal{Y}$ is a 'back off' function and the threshold $t \in \mathcal{T} \subseteq \mathbb{R}$ controls how much $F_t(x, y_0)$ 'backs off' from the base output y_0 by removing (unreliable) claims. We call F_t sound if it satisfies the property that $F_{\sup \mathcal{T}}(x, y_0) = \varnothing$, where $\varnothing$ represents some output sequence that abstains from making any claim. For



x = Who was Abe Lincoln?

$\mathsf{F}_1(x)$ = Abraham Lincoln, born in Idaho, was the 16th President of the United States. He is best known for leading the country through the Civil War.

$\mathsf{F}_2(x)$ = Abraham Lincoln was the 16th President of the United States. He is best known for leading the country through the Civil War.

$\mathsf{F}_3(x)$ = Abraham Lincoln was the 16th President of the United States.

$\mathsf{F}_4(x) = \varnothing$

Figure 6: An example of $F_t(x)$ for $t \in \mathcal{T}$.

notational clarity, we will omit the second argument whenever there is only one relevant language model $\mathcal{P}(x)$ that can generate y_0. In this case, we use the shorthand $F_t(x) := F_t(x, \mathcal{P}(x))$.

For the score function, we can redefine the score in (1) as

$$S(x, y^*) := \inf\{t \in \mathcal{T} : \forall j \geq t, y^* \in E(F_j(x))\}.$$

This matches the original score with one minor modification where we take the minimum *strictly safe* threshold—we consider a threshold *strictly safe* if any threshold greater than or equal to this one is safe. For the example in Figure 5, if we add $y_3 = \varnothing$ and define $F_t(x) := y_t$, we would have the minimum strictly safe threshold $r(x, y^*) = 2$.

With these two components in hand, we can directly apply the split conformal prediction method to obtain an LM with our desired correctness guarantees. Formally, we say that a model $\mathcal{P}$ is α-conformally factual if for exchangeable $(X_i, Y_i^*) \in \mathcal{X} \times \mathcal{Y}, i \in [n+1]$ and $\{(X_i, Y_i^*)\}_{i=1}^n$ used to construct $\widetilde{\mathcal{P}}$, the reference output Y_{n+1}^* satisfies the following inequality:

$$\mathbb{P}(Y_{n+1}^* \in E(\widetilde{\mathcal{P}}(X_{n+1}))) \geq 1 - \alpha.$$

The procedure can be described as follows.

1. For $1 \leq i \leq n$, define
$$S(X_i, Y_i^*) := \inf\{t \in \mathcal{T} : \forall j \geq t, Y_i^* \in E(F_j(X_i))\}.$$
2. Let q_α be the $\lceil (n+1)(1-\alpha) \rceil / n$th quantile of these scores $S(X_i, Y_i)$ for $i = 1, 2, \ldots, n$.
3. Output $F_{q_\alpha}(X_{n+1})$

Theorem. Suppose $\{(X_{n+1}, Y_{n+1}^*)\}$ are exchangeable, and F_t is sound. Then for $\alpha \geq 1/(n+1)$, we have

$$\mathbb{P}(Y_{n+1}^* \in E(F_{q_\alpha}(X_{n+1}))) \geq 1 - \alpha.$$

Original	**Conformal Factuality**
Zamfir Ralli-Arbore (1848-1933) was a Romanian political activist and historian from Bessarabia, who spent much of his life in exile. As a member of the National Liberal Party, he campaigned for the union of his native region with the Kingdom of Romania, and was a prominent opponent of Russian and Soviet policies. He was also a noted historian, specializing in the history of the Moldavia and Wallachia during the Middle Ages.	Zamfir Ralli-Arbore, born in 1848, was a Romanian political activist from Bessarabia. He passed away in 1933.
John Estes may refer to: 1. Sleepy John Estes (1899-1977), American blues guitarist, songwriter and vocalist. 2. John Adam Estes (1919-2001), American politician who served as a member of the Texas House of Representatives. 3. John Estes (American football) (born 1986), American football center. 4. John C. Estes, American screenwriter known for his work on films like "Mean Creek" and "The Details". Without more specific information, it's hard to provide details on the correct John Estes.	Sleepy John Estes was an American blues guitarist, songwriter and vocalist. He was born in 1899 and died in 1977.
Michael Valpy is a Canadian journalist, author, and lecturer. He was born on July 30, 1942 in Toronto, Ontario. He is best known for his work as a senior writer for The Globe and Mail, a leading Canadian newspaper, where he covered both national and international news. Valpy has also written several books on Canadian politics and history, and has taught journalism at several universities. He is a three-time nominee for the Governor General's Award, Canada's top literary honor.	Michael Valpy is a Canadian journalist, author, and lecturer. He is best known for his work as a senior writer for The Globe and Mail, a leading Canadian newspaper. In his career, Valpy covered both national and international news for the newspaper. In addition to his journalism career, he has written several books on Canadian politics and history. Valpy has also taught journalism at several universities.
We have $i^5 = i^4 \cdot i = i$, $i^{-25} = i^{-24} \cdot i^{-1} = i$, and $i^{45} = i^{44} \cdot i = i$. Therefore, $i^5 + i^{-25} + i^{45} = \boxed{3i}$	The given steps are:$i^5 = i^4 \cdot i = i$\n\n$i^{45} = i^{44} \cdot i = i$\n\nUsing these steps, we can simplify the given expression:$i^5 + i^{-25} + i^{45} = i + i^{-25} + i = i + i^{-25} + i$

Figure 7: Comparison of original output by GPT-4 versus our modified output from conformal factuality with frequency scoring.

Proof. By the exchangeability, we have

$$P(S(X_{n+1}, Y^*_{n+1}) \le q_\alpha) = \frac{\lceil (n+1)(1-\alpha) \rceil}{n+1} \ge 1 - \alpha.$$

Note that $S(X_{n+1}, Y^*_{n+1}) \le q_\alpha$ implies that $Y^*_{n+1} \in E(F_{q_\alpha}(X_{n+1}))$, i.e., $Y^*_{n+1} \implies F_{q_\alpha}(X_{n+1})$. Thus

$$\mathbb{P}(Y^*_{n+1} \implies F_{q_\alpha}(X_{n+1})) \ge 1 - \alpha.$$