Measuring the Impact of Lexical Training Data Coverage on Hallucination Detection in Large Language Models

Shuo Zhang[1], Fabrizio Gotti[1], Fengran Mo[1], Jian-Yun Nie[1]
[1]Université de Montréal, Québec, Canada
{shuo.zhang, fabrizio.gotti, fengran.mo}@umontreal.ca, nie@iro.umontreal.ca

Abstract

Hallucination in large language models (LLMs) is a fundamental challenge, particularly in open-domain question answering. Prior work attempts to detect hallucination with model-internal signals such as token-level entropy or generation consistency, while the connection between pretraining data *exposure* and hallucination is underexplored. Existing studies show that LLMs underperform on long-tail knowledge, i.e., the accuracy of the generated answer drops for the ground-truth entities that are rare in pretraining. However, examining whether data coverage itself can serve as a detection signal is overlooked. We propose a complementary question: *Does **lexical** training-data coverage of the question and/or generated answer provide additional signal for hallucination detection?* To investigate this, we construct scalable suffix arrays over RedPajama's 1.3-trillion-token pretraining corpus to retrieve n-gram statistics for both prompts and model generations. We evaluate their effectiveness for hallucination detection across three QA benchmarks. Our observations show that while occurrence-based features are weak predictors when used alone, they yield modest gains when combined with log-probabilities, particularly on datasets with higher intrinsic model uncertainty. These findings suggest that lexical coverage features provide a complementary signal for hallucination detection. All code and suffix-array infrastructure are provided at https://github.com/WWWonderer/ostd.

1 Introduction

Hallucination, where a language model produces factually incorrect or unsupported content, remains a central and unresolved challenge in building reliable generative systems (Huang et al., 2025). Prior work has attempted to detect and mitigate the hallucination issue along three major lines: (1) analyzing intrinsic uncertainty signals such as token-level log-probabilities and entropy (Kadavath et al., 2022; Kuhn et al., 2023); (2) evaluating generation faithfulness via self-consistency or sample agreement (Manakul et al., 2023; Wang et al., 2023); and (3) incorporating external knowledge or retrieval mechanisms to constrain generation (Shuster et al., 2021; Paudel et al., 2025).

From the perspective that LLMs act as statistical compressors of their training data (Delétang et al., 2023), it is intuitive to investigate the connection between pretraining data exposure and hallucination, which remains underexplored in the literature. Kandpal et al. (2023) shows that question answer (QA) accuracy declines for entities with limited presence in the pretraining corpus, evidencing a long-tail effect. This suggests that directly examining an LLM's pretraining data may provide useful signals for detecting hallucinations.

In this work, we investigate the predictive value of training-data exposure from a complementary, *lexical* perspective. We operationalize exposure via n-gram occurrence statistics in the model's pretraining corpus. Concretely, we compute tokenized n-gram frequencies ($1 \leq n \leq 5$) for both prompted questions and model-generated answers across three QA benchmarks, extracting counts with a suffix-array index over RedPajama's 1.3T-token corpus (Together.ai, 2023a). Unlike entity-linking-based approaches, which test the coverage assumption only for named entities and rely on external resources such as *Wikipedia* to identify them, we examine training-data coverage beyond entities, which are crucial for questions such as "*Who said 'I think, therefore I am?'*" or "*Which play features 'To be or not to be?'*".

Our hypothesis is that these surface-level statistics offer a complementary signal to those used in prior work. While learned embeddings encode distributed co-occurrence patterns across multiple training contexts and the entire vocabulary, raw n-gram occurrence statistics reflect the direct lexical overlap of specific token sequences. As

such, they may better capture novelty or out-of-distribution behavior, particularly when the model generates spans that were never explicitly seen during training. We test this hypothesis by evaluating occurrence-based features both independently and in combination with intrinsic model signals such as log-probabilities. Our results show that:

- Prompts with higher training data n-gram frequency are, on average, less likely to yield hallucinated generations in non-extractive QA. However, occurrence statistics alone remain noisy and weak predictors.
- In the extractive QA task, the frequency of a generated answer span in the training corpus is a strong indicator of truthfulness.
- When combined with output log-probabilities, occurrence features yield consistent improvements over log-probabilities alone, though the gains vary across datasets.

2 Related Work

Hallucination Detection. Hallucination detection methods generally fall into three categories: intrinsic, consistency-based, and retrieval-augmented approaches. Among intrinsic methods, Kadavath et al. (2022) showed that a model's confidence in its answer can be estimated by training a classifier on its output embeddings. SelfCheckGPT (Manakul et al., 2023) and the semantic entropy approach of Kuhn et al. (2023) further demonstrated that token-level log-probabilities and entropies can serve as simple yet effective signals for detecting hallucinations. SelfCheckGPT also introduced a consistency-based approach that estimates hallucination rates by measuring divergence across multiple model generations. Similarly, Wang et al. (2023) showed that incorporating chain-of-thought prompting with self-consistency can further reduce hallucinations. Other studies (Shuster et al., 2021; Paudel et al., 2025) use external knowledge sources or grounding documents to constrain generation and improve factuality. However, such resources may not always be available or reliable in practice.

Unlike prior approaches that rely on model-internal signals or external knowledge, Kandpal et al. (2023) show that a causal relationship exists between the presence of QA entities in an LLM's pretraining data and its ability to answer them, as LLMs struggle particularly with rarely seen topics. Their method, however, depends on large-scale entity extraction and linking to a curated catalog, introducing pipeline dependencies and potential extraction noise.

In contrast, we investigate a complementary, low-level, model-agnostic signal: the occurrence frequency of lexical spans in the pretraining corpus. This provides a more direct measure of a topic's coverage in the training data. To the best of our knowledge, this is the first systematic study that evaluates n-gram occurrence statistics as predictors of hallucination.

LLM Pretraining and Data Infrastructure. Our study builds on the RedPajama dataset and LLM family (Together.ai, 2023a,b), which replicates the training recipe of the original LLaMA (Touvron et al., 2023) using publicly available data. To extract training data occurrence statistics at scale, we use the suffix array indexing infrastructure introduced by Lee et al. (2022), originally developed to analyze memorization and deduplication in large corpora.

3 Methodology

3.1 Problem Setup

We frame hallucination detection as a binary classification task: given a prompt and its corresponding model-generated output, we predict whether the output is hallucinated. Ground-truth labels are assigned using one of the evaluation metrics with specific thresholds. Our goal is to evaluate whether occurrence-based features derived from the model's training data can help predict hallucination, either alone or in combination with intrinsic confidence features such as token log-probabilities.

3.2 Occurrence Extraction via Suffix Array

Efficient access to n-gram statistics requires a scalable indexing structure. Building on Lee et al. (2022), we adapt their Rust-based suffix array implementation by developing a minimal Rust–Python bridge using PyO3 (Hewitt et al., 2024), and integrate it into our Python pipeline for large-scale n-gram count extraction over RedPajama's 1.3T-token corpus.

We tokenize the raw corpus using the same tokenizer as the RedPajama model and flatten all sequences by concatenating them with the tokenizer's end-of-sequence (EOS) token. We then construct a suffix array over this flattened token sequence using the adapted infrastructure described above. The core indexing algorithm is SA-IS (Nong et al., 2009), which builds the suffix array in linear time

3-gram	wikipedia	arxiv	c4	cc_2019	cc_2020	cc_2021	cc_2022	cc_2023	github	sx
Published on Feb	0	0	710	2184	1693	1800	2533	3883	4	5
on Feb 21	15	12	2616	6476	5234	5581	4538	4402	27	22
21, 1848	96	0	353	705	837	713	936	1009	0	0
for the Communist	934	2	3723	8830	11457	10714	10020	10623	2	1
Manifesto	18970	357	124171	271742	353273	306758	261659	277228	1231	702

Table 1: Example 3-grams and their raw frequencies across the RedPajama training subsets. cc_2019–cc_2023 denote Common Crawl subsets by year. sx refers to StackExchange.

by exploiting local ascending and descending patterns in the data. Once constructed, the suffix array enables efficient substring and n-gram lookup in $O(\log m)$ query time, where m is the length of the corpus.

3.3 Log-Probability Features

We extract two intrinsic confidence features from the model: the average log-probability of the generation and the average log-probability of the prompt.

Generation log-probability. Following prior work (Manakul et al., 2023; Jesson et al., 2024), given a prompt $x = (x_1, \ldots, x_m)$ and a generation $y = (y_1, \ldots, y_n)$, the generation log-probability is computed as the average conditional log-probability of the output tokens conditioned on the prompt:

$$\text{Gen\_LogP}(y \mid x) = \frac{1}{n} \sum_{t=1}^{n} \log p(y_t \mid x, y_{<t}) \quad (1)$$

This reflects the model's confidence in its own output, as conditioned on the full prompt and preceding tokens.

Prompt log-probability. We also compute the average log-probability of the prompt itself, based on its left-to-right autoregressive likelihood. Specifically, given a prompt $x = (x_1, \ldots, x_m)$, we compute the following equation:

$$\text{Pr\_LogP}(x) = \frac{1}{m-1} \sum_{t=2}^{m} \log p(x_t \mid x_{<t}) \quad (2)$$

This measures how "typical" the prompt is under the model's own language distribution.

3.4 Occurrence-Based Features

We extract n-gram occurrence counts from the RedPajama training corpus (Together.ai, 2023a). For prompts, we compute average occurrence counts over n-grams of length 1 to 5. For generations, which are typically shorter in length, we compute average 1-gram and 2-gram occurrence counts. We also compute the occurrence counts of the entire generation if it appears verbatim in the training data. These counts are raw frequency values without normalization. Unless otherwise specified, stopwords are retained in the n-gram construction. We provide an example as follows to illustrate the 3-gram extraction procedure.

Example. Given the question: *"Published on Feb 21, 1848, which two authors were responsible for the Communist Manifesto?"*, we extract overlapping 3-grams such as: *"Published on Feb"*, *" on Feb 21"*, *" 21, 1848"*, *" for the Communist"*, *" Manifesto"*, etc. Note that the n-grams are based on subword tokens defined by the tokenizer, and may not align with word boundaries. For example, *" Manifesto"* is tokenized into three separate tokens as *" Man"*, *"if"*, and *"esto"*, and thus appears as a 3-gram even though it looks like a single word.

Each 3-gram token sequence is then queried against the suffix array index to retrieve its frequency across subsets of the RedPajama training corpus. Table 1 shows example count statistics in various collections. For each n-gram, we sum its raw frequencies across all available RedPajama subsets[1] to compute a total count, which is used in the modeling described in the next section.

In addition to fixed-length n-gram statistics, we extract occurrences of semantically meaningful key phrases from each prompt. These key phrases are generated using GPT-4o (OpenAI, 2024), which is instructed to return a list of canonical and paraphrased answer-seeking spans from the question. For instance, given the prompt in the example above, the extracted key phrases contain *"Communist Manifesto"*, *"authors of Communist Manifesto"*, etc. We query each key phrase against the suffix array and compute its total frequency across the training corpus, similar to the n-grams. These values serve as an additional feature reflect-

[1]We omit the book subset due to copyright-related restrictions. It comprises a small fraction of the full corpus.

ing how often the most answer-relevant spans in the prompt have been seen during training. To ensure robustness, we expand key phrases across different capitalization and spacing variants.

Raw Frequency Model. Our simplest modeling approach uses feature vectors built from average n-gram occurrence counts. For each prompt or generation, we extract all n-grams and key phrases, query their frequency in the training corpus, and compute the mean count score $S_{raw}(z)$:

$$S_{raw}(z) = \frac{1}{|G(z)|} \sum_{g \in G(z)} \text{Count}(g) \quad (3)$$

where z is the sequence from the prompt or the generation, $G(z)$ is the set of all extracted n-grams from z for a given n (or the set of extracted key phrases), and $\text{Count}(g)$ is the total frequency of an n-gram or keyphrase g from $G(z)$ in the training corpus (summed across all subsets).

We construct separate average features for prompt and generation n-grams across each n value, which are then used as inputs to downstream classifiers or plotted independently against hallucination rates.

Classic n-gram Scoring Model. In addition to average raw frequency, we implement a standard n-gram language model using conditional likelihoods. Specifically, for a sequence $z = (z_1, \ldots, z_T)$, we extract all n-grams and their corresponding $(n-1)$-gram prefixes, and define a log-likelihood-style score $S_{ng}(z)$:

$$S_{ng}(z) = \frac{1}{T-n+1} \sum_{t=1}^{T-n+1} \log \left(\frac{\text{Count}(g_t^n) + \epsilon}{\text{Count}(g_t^{n-1}) + \epsilon} \right) \quad (4)$$

where $\text{Count}(g_t^n)$ and $\text{Count}(g_t^{n-1})$ denote the corpus frequencies of the n-gram $(z_t \ldots z_{t+n-1})$ and its $(n-1)$-gram prefix $(z_t \ldots z_{t+n-2})$, respectively. The $\epsilon = 10^{-8}$ is a small constant for smoothing.

This score reflects how likely a given sequence z from the prompt or generation is under a classic count-based language model induced from the training data. It serves as a complementary signal to the prompt log-likelihood in Eq. 2, offering a data-driven measure of lexical familiarity. As with the raw frequency model, we use these scores as input features for downstream classification, and also analyze them independently in relation to hallucination rates.

Stopword Filtering. We apply two filtering strategies via the NLTK stopword list (Bird et al., 2009). For the *raw frequency model* (Eq. 3), we discard any n-gram g such that $\texttt{StopwordFrac}(g) = \frac{\#\text{stopwords in } g}{|g|} > 0.66$. For the *$n$-gram scoring model* (Eq. 4), we exclude n-grams whose final token is a stopword to avoid uninformative completions dominating likelihood scores.

4 Experimental Setup

4.1 Datasets

We evaluate our hallucination detection methods on three question answering (QA) benchmarks: **TriviaQA** (Joshi et al., 2017), **CoQA** (Reddy et al., 2019), and **NQ-Open** (Lee et al., 2019). TriviaQA and NQ-Open are both open-domain QA datasets in which models must generate free-form factual answers without access to grounding context. In contrast, CoQA is a conversational, extractive QA dataset where answers typically consist of short spans copied from a provided passage. These datasets differ in answer format, contextual grounding, and generation constraints, allowing us to evaluate the robustness of our methods across both open-ended and extractive QA settings.

4.2 Model and Generations

We use the RedPajama-INCITE language model family to generate answers for each question, including both the 3B and 7B size models. We chose RedPajama because it is one of the few high-quality, openly available LLMs for which both the model weights and the full training corpus (1.3T tokens) are publicly accessible. Furthermore, the scale of RedPajama's training set is large enough to be representative of modern LLMs, yet small enough to allow tractable indexing and querying using academic compute resources. All generations are produced in a few-shot setting, where the question is prepended with a small number of in-context examples in a standard `Q: / A:` format. For computing occurrence statistics, we isolate and use only the current test-time question. We use stochastic decoding with top-k sampling ($k = 50$), nucleus sampling ($p = 0.7$), and temperature set to 1.0.

We evaluate each generation by comparing it to the gold reference answers using two hallucination criteria: (1) **Exact Match (EM)**, where hallucination is defined as failure to exactly match any reference; and (2) **ROUGE-L** (Lin, 2004), where a generation is labeled hallucinated if its ROUGE-L

score is below 0.3, similar to Kuhn et al. (2023).

4.3 Evaluation and Metrics

We evaluate our extracted features in two stages: (1) by computing their standalone correlation with hallucination using the Area Under the Receiver Operating Characteristic (AUROC) curve, and (2) by testing their predictive value in downstream binary classifiers.

Feature Correlation via AUROC. To assess the utility of each feature independently, we compute the AUROC between its values and binary hallucination labels. A feature with AUROC > 0.5 indicates positive correlation with correct generations, while AUROC $= 0.5$ corresponds to random guessing. We perform this analysis across all three datasets under EM and ROUGE-L.

Classifier-Based Evaluation. We train supervised binary classifiers to distinguish hallucinated from non-hallucinated outputs using selected features.

Feature Sets. We compare two configurations:

- **Logprob only**: generation log-probability as a single feature.
- **Full feature set**: generation log-probability, prompt log-probability, and the best-performing prompt-side and generation-side occurrence features identified via AUROC.

All features are standardized with `StandardScaler` from `scikit-learn` (Pedregosa et al., 2011).

Model Types. We evaluate two classifier architectures:

- **Decision Tree**: shallow trees (depth 3–20) trained with `scikit-learn`, offering interpretable decision boundaries and feature importances.
- **Neural Network (MLP)**: a 3-layer MLP (32–64–32, ReLU activations, softmax output) trained with Adam (lr $= 10^{-4}$, batch size 20) for 25 epochs. Architectural variations had negligible effect on observed trends.

Additionally, for the **Logprob only** feature set, we train a logistic regression baseline to learn a single decision threshold.

Training Setup. To control for differences in hallucination rates across datasets, we construct balanced 50/50 splits for each dataset by sampling equal numbers of hallucinated and non-hallucinated examples from each model's generated outputs, using either ROUGE-L or EM as the labeling criterion. Generations with fewer than two tokens are excluded, as they yield undefined 2-grams in the n-gram model and typically correspond to empty or trivial outputs. We train all classifiers using an 80/20 train–test split per dataset, repeating each configuration with five random seeds and reporting mean accuracy and standard deviation across runs.

Evaluation Metric. We report test-set accuracy as the main evaluation metric.

5 Experimental Results

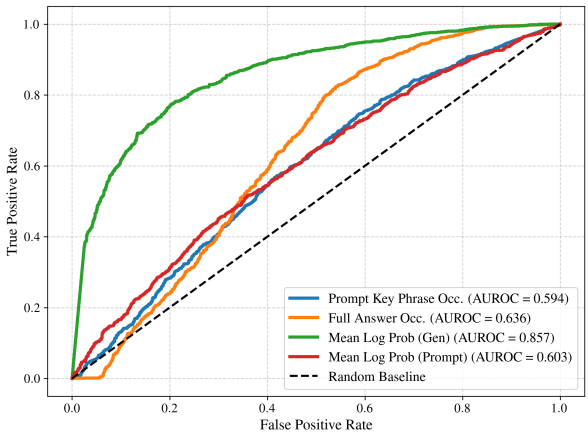
We report the 7B model under the EM criterion and display the best-performing AUROC curves from each category. The full results are provided in Appendix A.1.

5.1 Feature-Level Correlation (AUROC)

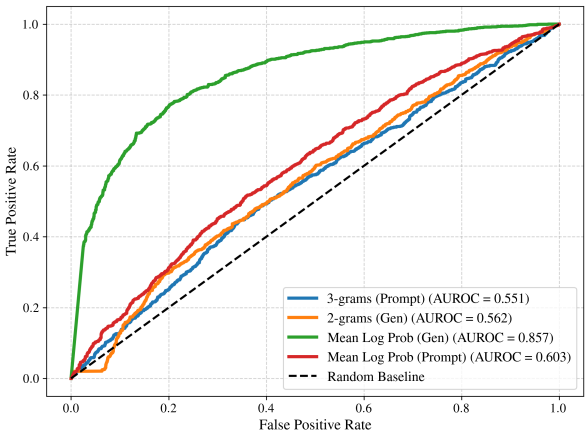
We first assess the standalone predictive value of log-probabilities and occurrence-based features by computing AUROC against hallucination labels. Figure 1 presents AUROC curves across the three datasets using raw frequency features (S_{raw}) shown on the left and n-gram features (S_{ng}) on the right. We observe that generation token log-probabilities consistently outperform all other features in AUROC, indicating they are the most reliable standalone predictor of hallucination. In contrast, occurrence-based features yield weaker and more variable signals. For prompt-side signals, we observe a modest positive association with model faithfulness (AUROC ≈ 0.6) in open-domain QA, but no measurable correlation in extractive QA. For generation-side signals, we typically observe an S-shaped trend: highly frequent generations are often hallucinated, reflecting memorized or templated patterns, whereas less frequent ones below a certain threshold tend to represent genuine entities, showing a positive correlation with faithfulness. In addition, we observe a strong correlation between the corpus occurrence of the extracted answer span and faithfulness under the EM evaluation setting in extractive QA.

5.2 Classifier Performance

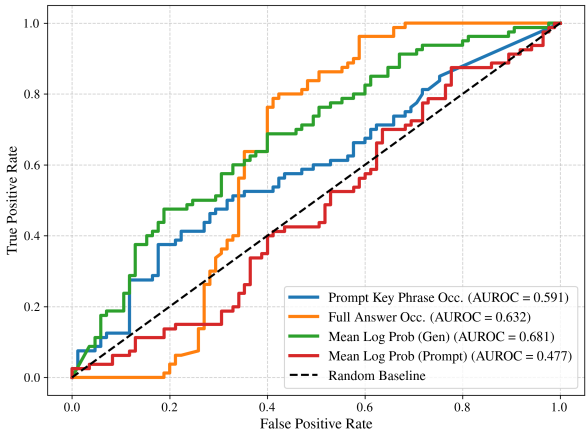
To evaluate the joint predictive power of log-probabilities and occurrence-based features, we train decision trees and neural networks to classify hallucinations using both logprob-only and



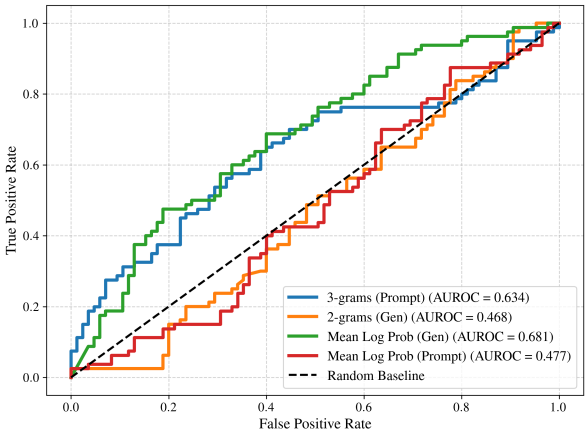
(a) TriviaQA (Raw Frequency)



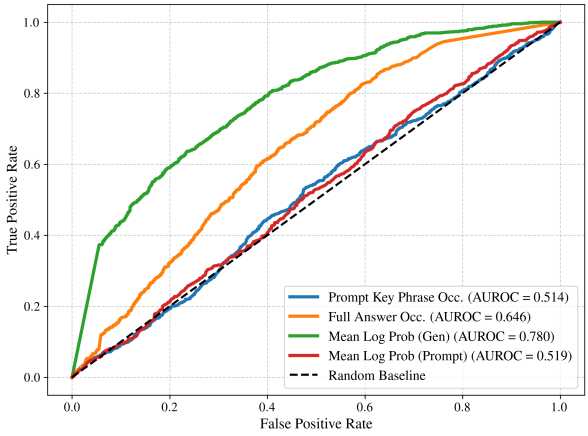
(b) TriviaQA (n-gram Model)



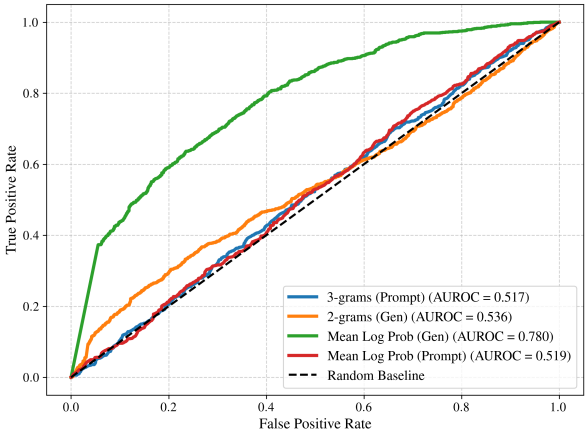
(c) NQ-Open (Raw Frequency)



(d) NQ-Open (n-gram Model)



(e) CoQA (Raw Frequency)



(f) CoQA (n-gram Model)

Figure 1: AUROC curves comparing log-probabilities and occurrence-based features across datasets with RedPajama-INCITE-7B model and EM evaluation.

full-feature input settings. For the raw frequency model, the most effective occurrence-based features are often the average prompt key phrase frequency and the full-generation frequency. For the n-gram model, longer n-grams such as 3-grams or 4-grams typically outperform shorter ones, while 5-grams tend to perform worse due to data sparsity, i.e., the exact 5-gram often does not appear in the training corpus. Table 2 reports test accuracies across model types, averaged over five random seeds. Across datasets and feature types, incorporating occurrence-based features generally yields the best results, with the strongest improvements observed on NQ-Open, where questions are more

Dataset	Model	Depth	LogProb	Full (Raw Freq)	Full (N-gram)
TriviaQA	*Threshold*	–	0.762 ± 0.018	–	–
	Tree	3	0.773 ± 0.026	$\mathbf{0.781} \pm 0.026$	0.772 ± 0.025
	Tree	7	0.758 ± 0.017	0.778 ± 0.014	0.759 ± 0.025
	NN	–	0.776 ± 0.023	$\mathbf{0.781} \pm 0.018$	0.771 ± 0.016
NQ-Open	*Threshold*	–	0.642 ± 0.126	–	–
	Tree	3	0.588 ± 0.076	0.727 ± 0.057	0.552 ± 0.141
	Tree	7	0.552 ± 0.089	$\mathbf{0.733} \pm 0.066$	0.612 ± 0.112
	NN	–	0.546 ± 0.111	0.588 ± 0.102	0.582 ± 0.136
CoQA	*Threshold*	–	0.697 ± 0.012	–	–
	Tree	3	0.712 ± 0.012	$\mathbf{0.728} \pm 0.010$	0.712 ± 0.008
	Tree	7	0.701 ± 0.020	0.705 ± 0.015	0.708 ± 0.017
	NN	–	0.707 ± 0.013	0.704 ± 0.020	0.714 ± 0.012

Table 2: Test accuracy of classifiers using the logprob-only and full feature sets. Values are mean $\pm$ standard deviation over 5 runs. Bold indicates the best result per dataset.

challenging and the log-probability signal alone is less reliable. A similar trend is observed under ROUGE-L evaluation, with full results provided in Appendix A.2.

5.3 Effect of Stopword Filtering

We applied stopword filtering to prompt-based features (see Section 3) and observed slight AUROC improvements, particularly for shorter n-grams. Classifier accuracy also improved modestly on some datasets, especially TriviaQA. Overall, however, the impact of stopword filtering was limited and did not materially change our conclusions. Full results are provided in Appendix A.3.

6 Analysis

6.1 Interpreting Occurrence Feature Utility

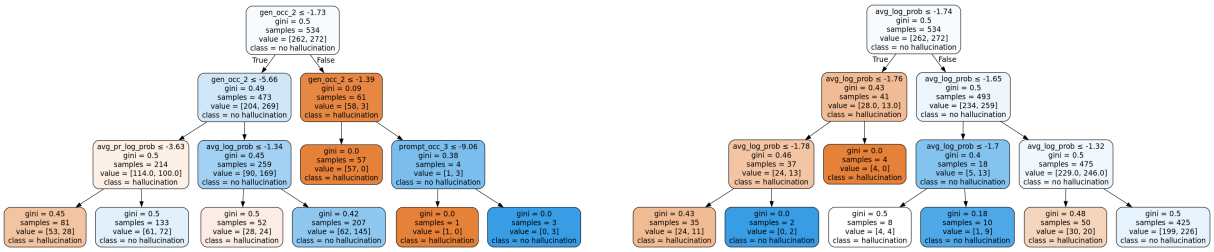
To understand why the full-feature decision tree achieves higher test accuracy than the log-probability-only counterpart despite similar training performance, we conduct a bootstrap resampling analysis on NQ-Open using the EM criterion. We repeatedly train 200 depth-3 trees on resampled versions of the training data and measure the fraction of test samples that receive identical predictions across all runs (*prediction consistency*). The log-probability-only model shows **zero consistency (0.00)**, indicating strong sensitivity to training perturbations, whereas the full-feature model achieves a **higher consistency of 0.06**. This improvement, though moderate in magnitude, reflects a clear directional trend: occurrence-based features lead to more reproducible decision boundaries and improved generalization stability.

We also identify concrete cases where occurrence features effectively differentiate hallucinated outputs. Specifically, surface artifacts such as repeated training prompts (e.g., `Q:`) in place of an actual answer. To examine how these distinctions arise, we visualize depth-3 decision trees trained with both the full feature set and log-probabilities alone. As shown in Figure 2, the tree with occurrence features cleanly separates 57 hallucinated instances along the path *gen_occ_2 (generation 2-gram score)* $> -1.73 \rightarrow$ *gen_occ_2* ≤ -1.39. Examples from this branch are presented in Table 3.

This split captures outputs with potentially high likelihood but low lexical diversity, particularly sequences that reproduce input-style prompts rather than providing substantive answers. Such artifacts cluster within a narrow 2-gram occurrence range under the pretraining distribution, enabling occurrence-based models to flag them more consistently. These examples demonstrate that training-data-aligned features complement intrinsic uncertainty measures by revealing brittle, format-driven hallucinations that token-level probabilities alone fail to detect.

6.2 Are Occurrence-Based Features Pure Noise?

As shown in Figure 1, occurrence-based features generally exhibit lower AUROC scores than intrinsic model signals. To assess whether they encode any meaningful signal rather than pure noise, we visualize the distribution of prompt 3-gram occurrence scores in TriviaQA (Figure 3). The two classes–hallucinated and non-hallucinated questions–show substantial overlap yet differ modestly in their means. We ran a t-test confirming the difference is statistically significant ($t = -4.21$, $p = 2.7 \times 10^{-5}$), though given the large sam-



(a) Full features (b) Log-probability only

Figure 2: Decision trees (depth 3) on NQ-Open. The full-feature model (left) splits early on generation 2-gram score (gen_occ_2), isolating a hallucination-prone cluster. The log-only tree (right) lacks this separation.

Question	Generation	Answer	LogP	2-gram score (gen)
When was the Tower of London finished being built	Q:	1078	-0.97	-1.42
How many episodes are there in Season 6 of Nashville	Q:	16	-0.38	-1.42
What is the revolution period of Venus in Earth years	Q:	224.7 Earth days / 0.615 yr	-1.29	-1.42

Table 3: Examples of hallucinated generations in NQ-Open where the model outputs Q:. These cases fall within a specific range of generation 2-gram scores (gen_occ_2) and are captured by an early split in the decision tree.

ple size, such tests are sensitive to even small mean shifts. This confirms that occurrence features capture a weak but consistent bias associated with hallucinated answers. While their entanglement with data frequency and format limits their standalone predictive power, they may nonetheless serve as complementary priors when combined with stronger indicators such as log-probability, as observed in Section 5.2.

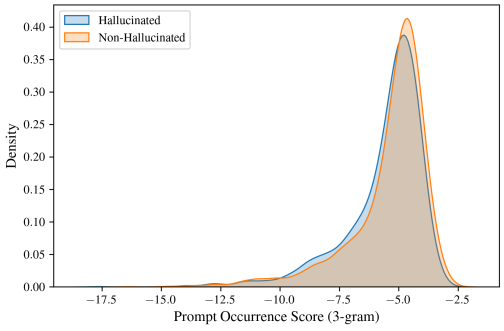


Figure 3: Distribution of prompt 3-gram scores for hallucinated and non-hallucinated answers on TriviaQA.

6.3 LLMs Are More Than Sequence Predictors: Evidence from Data Sparsity

LLMs are often viewed as large-scale compressors of world knowledge, retrieving content memorized from training data. Our findings suggest that this view is overly reductive. Empirically, we observe that even simple natural language questions often contain 4-grams or longer spans entirely absent from a 1.3T-token pretraining corpus. For instance, 21.1% of 4-grams and 44.4% of 5-grams in NQ-Open do not appear anywhere in the training data. Many missing n-grams are not semantically obscure; for example, "*asked king to send him*". Table 4 reports the percentage of missing n-grams across different test datasets. Detailed examples are provided in Appendix A.5. This indicates that models are frequently exposed to sequences they have never explicitly encountered during training, and that LLMs must generalize compositionally rather than relying solely on sequence memorization much more often than is commonly assumed.

Dataset	1-gram	2-gram	3-gram	4-gram	5-gram
TriviaQA	0.00	0.03	3.33	14.97	31.99
NQ-Open	0.00	0.06	4.79	21.09	44.44
CoQA	0.00	0.02	1.56	9.40	24.80

Table 4: Percentage of n-grams with zero occurrences in the RedPajama training corpus across question sets from different datasets.

7 Conclusion

We present, to our knowledge, the first systematic study of how lexical training data coverage, measured via n-gram frequency, relates to hallucination in LLMs. Our AUROC and classifier analyses show that while coverage-based features are weak standalone predictors, they can complement log-probabilities, especially when intrinsic confidence is unreliable. These results indicate that lexical-coverage features provide an additional, data-aware signal for hallucination detection tied to training exposure.

Limitations

Due to resource constraints, our findings are based on experiments with the RedPajama-INCITE models, and may not fully generalize to larger models with more sophisticated pretraining and post-training alignments. The suffix array used to compute occurrence statistics covers 1.3 trillion tokens, which, while extensive, does not reflect the full pretraining corpora used in proprietary models. Additionally, our analysis is limited to exact n-gram frequency counts and does not account for paraphrasing, synonymy, or semantic similarity. Finally, our results are correlational and do not establish a causal relationship between training data exposure and hallucination behavior.

Ethics Statement

This work studies hallucination detection in LLMs by analyzing training data coverage using n-gram statistics from RedPajama's openly released 1.3T-token pretraining corpus. While ethical concerns around the large-scale use of human-authored content in LLM training are real and ongoing, particularly regarding consent, compensation, and the displacement of knowledge workers, we note that RedPajama represents a relatively transparent and responsible effort, with clear documentation and the removal of copyrighted subsets such as books. We encourage the community to continue developing tools and norms for attribution, data auditing, and fair recognition of creators whose work fuels modern LLMs.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media, Inc.

Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, and 1 others. 2023. Language modeling is compression. *arXiv preprint arXiv:2309.10668*.

David Hewitt and 1 others. 2024. Pyo3: Rust bindings for python. https://github.com/PyO3/pyo3.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Andrew Jesson, Nicolas Beltran Velez, Quentin Chu, Sweta Karlekar, Jannik Kossen, Yarin Gal, John P Cunningham, and David Blei. 2024. Estimating the hallucination rate of generative ai. *Advances in Neural Information Processing Systems*, 37:31154–31201.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International conference on machine learning*, pages 15696–15707. PMLR.

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*.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Ge Nong, Sen Zhang, and Wai Hong Chan. 2009. Linear suffix array construction by almost pure induced-sorting. In *2009 data compression conference*, pages 193–202. IEEE.

OpenAI. 2024. Gpt-4o system card. https://arxiv.org/abs/2410.21276.

Bibek Paudel, Alexander Lyzhov, Preetam Joshi, and Puneet Anand. 2025. Hallucinot: Hallucination detection through context and common knowledge verification. *arXiv preprint arXiv:2504.07069*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803.

Together.ai. 2023a. Redpajama: An open source recipe to reproduce llama training dataset. https://www.together.ai/blog/redpajama.

Together.ai. 2023b. Releasing 3b and 7b redpajama-incite family of models including base, instruction-tuned & chat models. https://www.together.ai/blog/redpajama-models-v1.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

A Appendix

A.1 Full EM 7B Results

Figure 4 shows the complete AUROC results, and Table 5 presents the corresponding classification results for the RedPajama-7B model using EM as the evaluation metric.

A.2 Effects of Metric Used

Figure 5 reports the AUROC scores of raw frequency and n-gram features under ROUGE-L based labeling. The overall trends are consistent with those observed under EM, indicating that the effects of occurrence-based features are largely metric-independent. However, in extractive QA, the correlation between occurrence and faithfulness becomes noticeably weaker. Table 6 presents the corresponding classifier accuracy results, which follow a pattern similar to the EM based setting.

A.3 Effects of Stopword Filtering

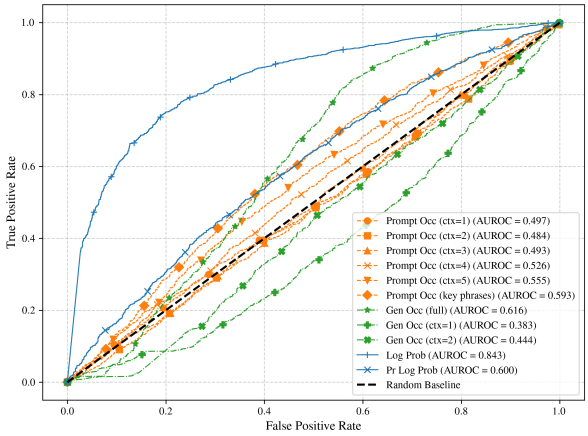
Figure 6 shows the AUROC scores of raw frequency and n-gram features after applying stopword filtering on prompts, as described in Section 3. We observe slight improvements in AUROC, particularly for shorter n-grams. Table 7 presents the corresponding classifier accuracy results. Filtering leads to modest test accuracy gains, especially on TriviaQA. However, the overall impact of stopword filtering is limited and does not significantly alter our core findings.

A.4 Effects of Model Size

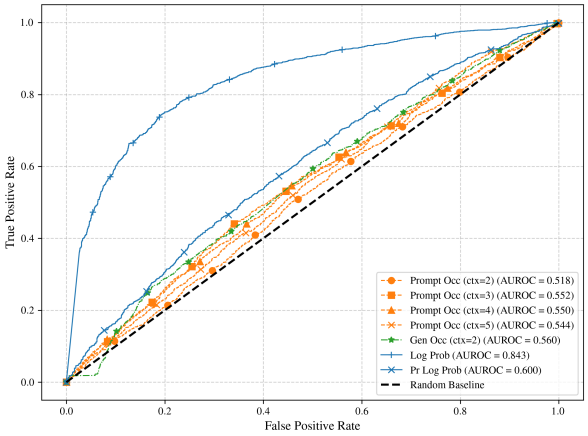
Figure 7 shows the AUROC scores of raw frequency and n-gram features under the RedPajama-INCITE-3B model. We observe similar overall trends to the 7B model. Table 8 presents the corresponding classifier accuracy results.

A.5 Data Sparsity Analysis

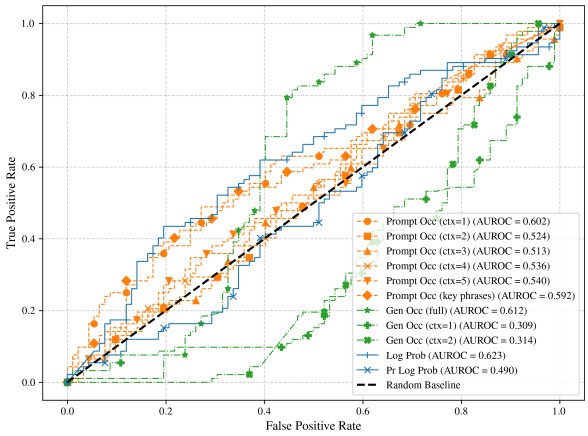
Table 9 reports the proportion of n-grams missing from the RedPajama training corpus across different test datasets, along with representative examples. Overall, a substantial proportion of 4-grams and longer sequences are absent verbatim from the training data, particularly in open-domain questions.



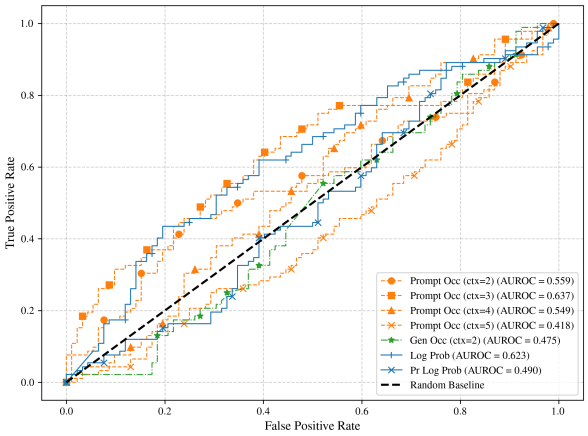
(a) TriviaQA (Raw Frequency)



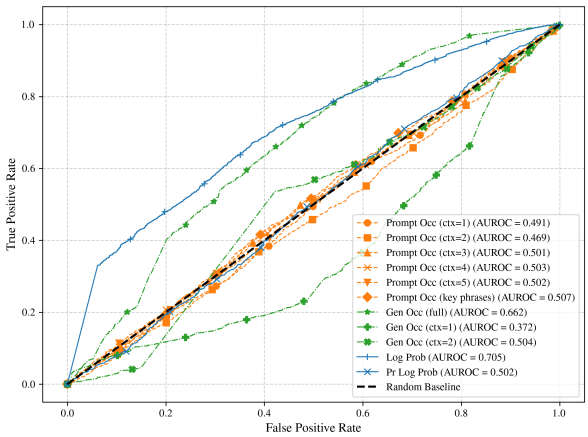
(b) TriviaQA (n-gram Model)



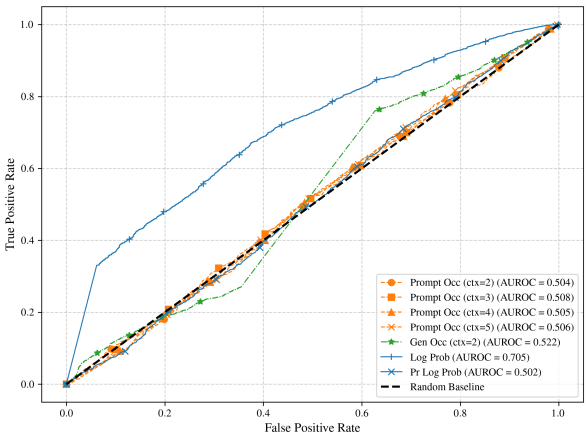
(c) NQ-Open (Raw Frequency)



(d) NQ-Open (n-gram Model)



(e) CoQA (Raw Frequency)

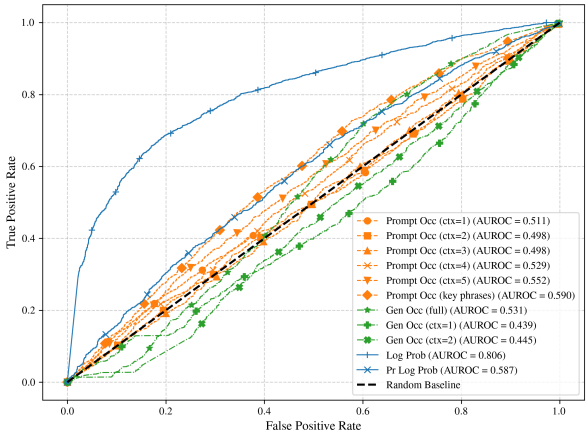


(f) CoQA (n-gram Model)

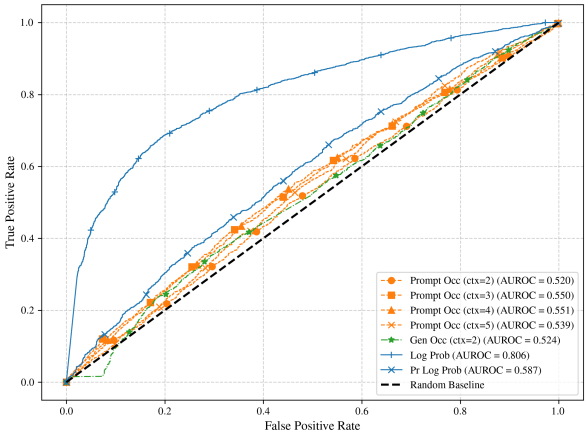
Figure 4: AUROC curves comparing log-probabilities and occurrence-based features across datasets. Model: RedPajama-INCITE-7B; metric: EM

Feature Type	Dataset	Model	Tree Depth	Train Acc (Log)	Train Acc (Full)	Test Acc (Log)	Test Acc (Full)
		Threshold	–	*0.775 ± 0.004*	–	*0.762 ± 0.018*	–
			3	0.787 ± 0.004	**0.796** ± 0.003	0.773 ± 0.026	**0.781** ± 0.026
			5	0.799 ± 0.005	**0.816** ± 0.002	0.766 ± 0.017	**0.782** ± 0.021
			7	0.810 ± 0.005	**0.843** ± 0.004	0.758 ± 0.017	**0.778** ± 0.014
	TriviaQA	Tree	9	0.830 ± 0.008	**0.882** ± 0.008	0.757 ± 0.022	**0.771** ± 0.015
			10	0.842 ± 0.009	**0.902** ± 0.006	0.756 ± 0.029	**0.762** ± 0.013
			20	0.945 ± 0.019	**0.998** ± 0.002	0.725 ± 0.027	**0.734** ± 0.021
		NN	–	0.786 ± 0.004	**0.791** ± 0.004	0.776 ± 0.023	**0.781** ± 0.018
		Threshold	–	*0.614 ± 0.033*	–	*0.642 ± 0.126*	–
			3	0.668 ± 0.020	**0.836** ± 0.016	0.588 ± 0.076	**0.727** ± 0.057
			5	0.726 ± 0.029	**0.892** ± 0.019	0.600 ± 0.072	**0.739** ± 0.059
			7	0.806 ± 0.032	**0.953** ± 0.020	0.552 ± 0.089	**0.733** ± 0.066
Raw Freq	NQ-Open	Tree	9	0.859 ± 0.028	**0.976** ± 0.012	0.552 ± 0.076	**0.727** ± 0.061
			10	0.870 ± 0.032	**0.983** ± 0.012	0.570 ± 0.087	**0.752** ± 0.069
			20	0.965 ± 0.014	**1.000** ± 0.000	0.570 ± 0.112	**0.727** ± 0.071
		NN	–	0.615 ± 0.053	**0.676** ± 0.065	0.546 ± 0.111	**0.588** ± 0.102
		Threshold	–	*0.700 ± 0.004*	–	*0.697 ± 0.012*	–
			3	0.718 ± 0.006	**0.738** ± 0.007	0.712 ± 0.012	**0.728** ± 0.010
			5	0.725 ± 0.004	**0.766** ± 0.003	0.709 ± 0.012	**0.724** ± 0.021
			7	0.736 ± 0.006	**0.797** ± 0.006	0.701 ± 0.020	**0.705** ± 0.015
	CoQA	Tree	9	0.751 ± 0.009	**0.843** ± 0.010	**0.702** ± 0.019	0.696 ± 0.026
			10	0.760 ± 0.010	**0.866** ± 0.013	**0.694** ± 0.020	0.688 ± 0.025
			20	0.864 ± 0.010	**0.991** ± 0.008	**0.677** ± 0.006	0.649 ± 0.024
		NN	–	0.711 ± 0.003	**0.718** ± 0.004	**0.707** ± 0.013	0.704 ± 0.020
		Threshold	–	*0.775 ± 0.004*	–	*0.762 ± 0.018*	–
			3	0.787 ± 0.004	**0.790** ± 0.002	**0.773** ± 0.026	0.772 ± 0.025
			5	0.799 ± 0.005	**0.808** ± 0.005	**0.766** ± 0.017	0.761 ± 0.034
			7	0.810 ± 0.005	**0.835** ± 0.006	0.758 ± 0.017	**0.759** ± 0.025
	TriviaQA	Tree	9	0.830 ± 0.008	**0.873** ± 0.009	**0.757** ± 0.022	0.752 ± 0.024
			10	0.842 ± 0.009	**0.891** ± 0.012	**0.756** ± 0.029	0.747 ± 0.022
			20	0.945 ± 0.019	**0.993** ± 0.004	**0.725** ± 0.027	0.719 ± 0.017
		NN	–	0.786 ± 0.004	**0.788** ± 0.006	**0.776** ± 0.023	0.771 ± 0.016
		Threshold	–	*0.614 ± 0.033*	–	*0.642 ± 0.126*	–
			3	0.668 ± 0.020	**0.770** ± 0.021	**0.588** ± 0.076	0.552 ± 0.141
			5	0.726 ± 0.029	**0.874** ± 0.036	0.600 ± 0.072	**0.612** ± 0.135
			7	0.806 ± 0.032	**0.939** ± 0.046	0.552 ± 0.089	**0.612** ± 0.112
N-gram	NQ-Open	Tree	9	0.859 ± 0.028	**0.977** ± 0.027	0.552 ± 0.076	**0.618** ± 0.106
			10	0.870 ± 0.032	**0.991** ± 0.016	0.570 ± 0.087	**0.594** ± 0.137
			20	0.965 ± 0.014	**1.000** ± 0.000	0.570 ± 0.112	**0.594** ± 0.143
		NN	–	0.615 ± 0.053	**0.623** ± 0.048	0.546 ± 0.111	**0.582** ± 0.136
		Threshold	–	*0.700 ± 0.004*	–	*0.697 ± 0.012*	–
			3	0.718 ± 0.006	**0.724** ± 0.005	0.712 ± 0.012	0.712 ± 0.008
			5	0.725 ± 0.004	**0.746** ± 0.008	0.709 ± 0.012	**0.720** ± 0.015
			7	0.736 ± 0.006	**0.784** ± 0.009	0.701 ± 0.020	**0.708** ± 0.017
	CoQA	Tree	9	0.751 ± 0.009	**0.827** ± 0.011	**0.702** ± 0.019	0.694 ± 0.005
			10	0.760 ± 0.010	**0.849** ± 0.013	**0.694** ± 0.020	0.691 ± 0.012
			20	0.864 ± 0.010	**0.985** ± 0.009	**0.677** ± 0.006	0.648 ± 0.008
		NN	–	0.711 ± 0.003	**0.721** ± 0.003	0.707 ± 0.013	**0.714** ± 0.012

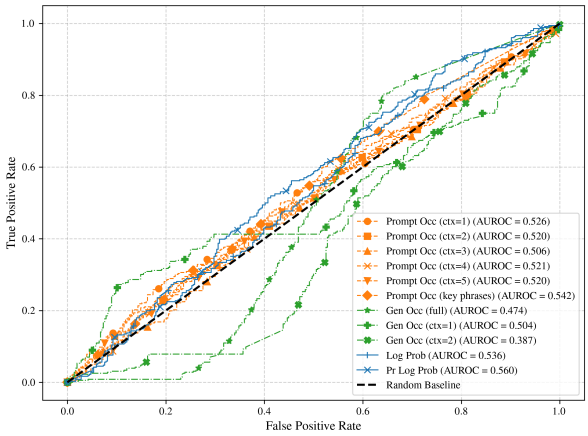
Table 5: Accuracy across tree depths, datasets, and feature types. Bold indicates higher accuracy between log-only and full features. Model: RedPajama-INCITE-7B; metric: EM.



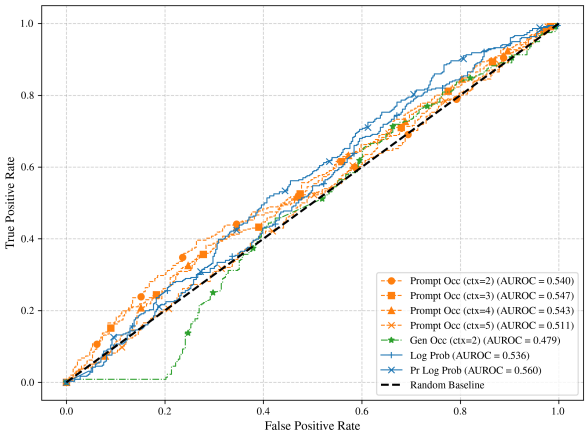
(a) TriviaQA (Raw Frequency)



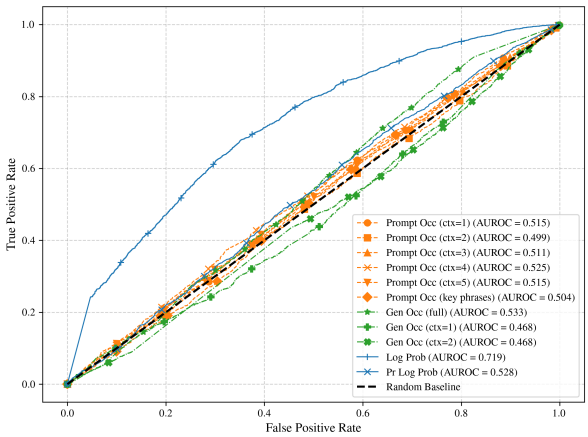
(b) TriviaQA (n-gram Model)



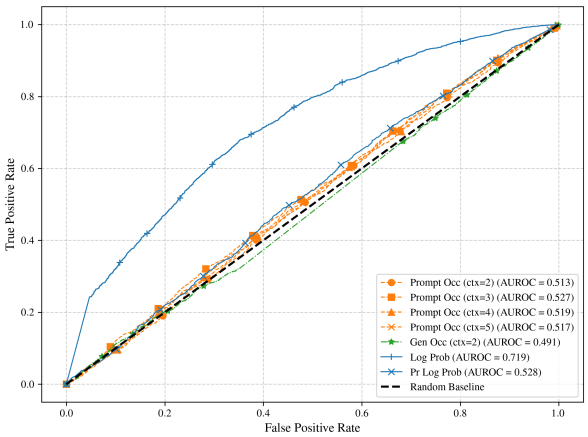
(c) NQ-Open (Raw Frequency)



(d) NQ-Open (n-gram Model)



(e) CoQA (Raw Frequency)

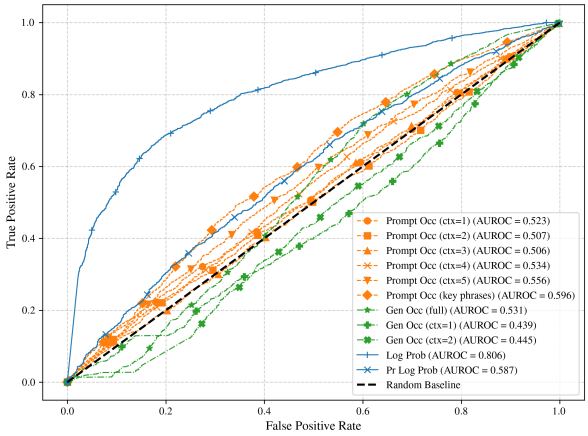


(f) CoQA (n-gram Model)

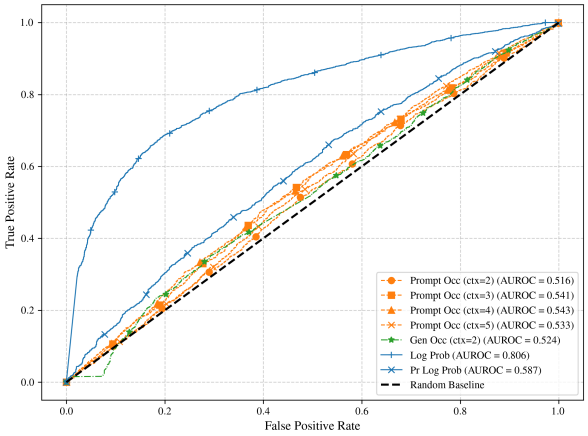
Figure 5: AUROC curves comparing log-probabilities and occurrence-based features across datasets. Model: RedPajama-INCITE-7B; metric: ROUGE-L.

Feature Type	Dataset	Model	Tree Depth	Train Acc (Log)	Train Acc (Full)	Test Acc (Log)	Test Acc (Full)
Raw Freq	TriviaQA	*Threshold*	–	*0.728 ± 0.003*	–	*0.731 ± 0.012*	–
		Tree	3	0.748 ± 0.004	**0.752** ± 0.003	0.744 ± 0.015	**0.750** ± 0.012
			5	0.753 ± 0.004	**0.762** ± 0.004	0.740 ± 0.013	**0.748** ± 0.007
			7	0.764 ± 0.002	**0.780** ± 0.004	0.734 ± 0.015	**0.741** ± 0.002
			9	0.777 ± 0.005	**0.806** ± 0.005	**0.733** ± 0.012	0.732 ± 0.004
			10	0.786 ± 0.007	**0.824** ± 0.008	**0.730** ± 0.010	0.724 ± 0.007
			20	0.892 ± 0.020	**0.973** ± 0.016	**0.699** ± 0.002	0.687 ± 0.005
		NN	–	0.746 ± 0.003	**0.752** ± 0.003	0.745 ± 0.013	**0.753** ± 0.013
	NQ-Open	*Threshold*	–	*0.536 ± 0.013*	–	*0.509 ± 0.045*	–
		Tree	3	0.566 ± 0.008	**0.682** ± 0.015	0.501 ± 0.045	**0.610** ± 0.037
			5	0.603 ± 0.017	**0.764** ± 0.007	0.521 ± 0.026	**0.652** ± 0.023
			7	0.651 ± 0.014	**0.844** ± 0.014	0.509 ± 0.035	**0.636** ± 0.032
			9	0.688 ± 0.017	**0.930** ± 0.005	0.487 ± 0.045	**0.642** ± 0.015
			10	0.710 ± 0.019	**0.956** ± 0.004	0.504 ± 0.044	**0.625** ± 0.027
			20	0.883 ± 0.045	**1.000** ± 0.000	0.501 ± 0.048	**0.630** ± 0.015
		NN	–	0.539 ± 0.009	**0.602** ± 0.015	0.518 ± 0.022	**0.548** ± 0.056
	CoQA	*Threshold*	–	*0.657 ± 0.003*	–	*0.654 ± 0.014*	–
		Tree	3	0.662 ± 0.003	**0.674** ± 0.002	0.649 ± 0.014	**0.669** ± 0.009
			5	0.668 ± 0.003	**0.687** ± 0.005	0.645 ± 0.011	**0.657** ± 0.014
			7	0.680 ± 0.004	**0.709** ± 0.003	**0.643** ± 0.012	0.640 ± 0.007
			9	0.693 ± 0.006	**0.746** ± 0.003	**0.639** ± 0.009	0.635 ± 0.019
			10	0.701 ± 0.009	**0.766** ± 0.007	**0.640** ± 0.013	0.617 ± 0.024
			20	0.821 ± 0.023	**0.949** ± 0.005	**0.612** ± 0.013	0.573 ± 0.015
		NN	–	0.658 ± 0.003	**0.660** ± 0.003	0.653 ± 0.016	**0.656** ± 0.015
N-gram	TriviaQA	*Threshold*	–	*0.728 ± 0.003*	–	*0.731 ± 0.012*	–
		Tree	3	0.748 ± 0.004	**0.751** ± 0.003	0.745 ± 0.015	**0.748** ± 0.012
			5	0.754 ± 0.004	**0.765** ± 0.005	0.740 ± 0.013	**0.746** ± 0.005
			7	0.764 ± 0.003	**0.785** ± 0.003	0.734 ± 0.015	**0.742** ± 0.007
			9	0.777 ± 0.005	**0.812** ± 0.003	0.733 ± 0.012	**0.735** ± 0.007
			10	0.786 ± 0.007	**0.826** ± 0.006	0.730 ± 0.010	**0.733** ± 0.007
			20	0.892 ± 0.020	**0.975** ± 0.013	**0.699** ± 0.002	0.694 ± 0.015
		NN	–	0.746 ± 0.003	**0.748** ± 0.005	0.745 ± 0.013	**0.750** ± 0.014
	NQ-Open	*Threshold*	–	*0.536 ± 0.013*	–	*0.509 ± 0.045*	–
		Tree	3	0.566 ± 0.008	**0.661** ± 0.013	0.502 ± 0.045	**0.593** ± 0.029
			5	0.603 ± 0.017	**0.708** ± 0.016	0.521 ± 0.026	**0.610** ± 0.040
			7	0.651 ± 0.014	**0.758** ± 0.025	0.509 ± 0.035	**0.613** ± 0.031
			9	0.688 ± 0.017	**0.815** ± 0.030	0.487 ± 0.045	**0.582** ± 0.038
			10	0.710 ± 0.019	**0.843** ± 0.032	0.505 ± 0.044	**0.581** ± 0.039
			20	0.883 ± 0.045	**0.979** ± 0.016	0.502 ± 0.048	**0.576** ± 0.038
		NN	–	0.539 ± 0.009	**0.611** ± 0.011	0.518 ± 0.022	**0.588** ± 0.031
	CoQA	*Threshold*	–	*0.657 ± 0.003*	–	*0.654 ± 0.014*	–
		Tree	3	0.662 ± 0.003	**0.664** ± 0.003	**0.649** ± 0.014	0.647 ± 0.011
			5	0.668 ± 0.003	**0.677** ± 0.005	**0.645** ± 0.011	0.634 ± 0.010
			7	0.680 ± 0.004	**0.700** ± 0.007	**0.643** ± 0.012	0.640 ± 0.012
			9	0.693 ± 0.006	**0.737** ± 0.016	**0.639** ± 0.009	0.633 ± 0.018
			10	0.701 ± 0.009	**0.759** ± 0.019	**0.640** ± 0.013	0.627 ± 0.013
			20	0.821 ± 0.023	**0.954** ± 0.019	**0.612** ± 0.013	0.589 ± 0.014
		NN	–	0.658 ± 0.003	**0.663** ± 0.004	0.653 ± 0.016	**0.660** ± 0.015

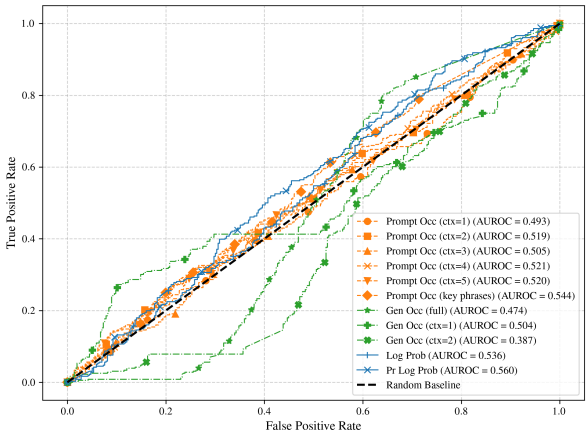
Table 6: Accuracy across tree depths, datasets, and feature types. Bold indicates higher accuracy between log-only and full features. Model: RedPajama-INCITE-7B; metric: ROUGE-L.



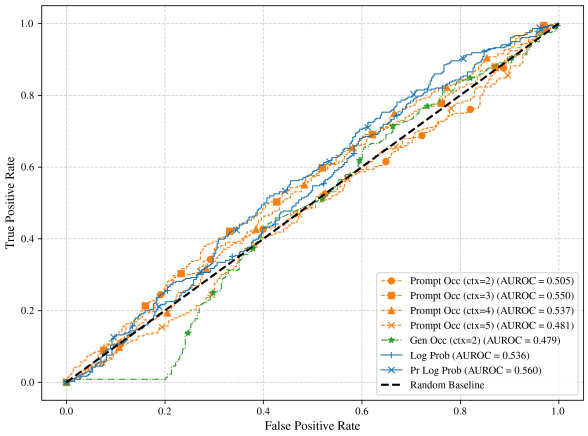
(a) TriviaQA (Raw Frequency)



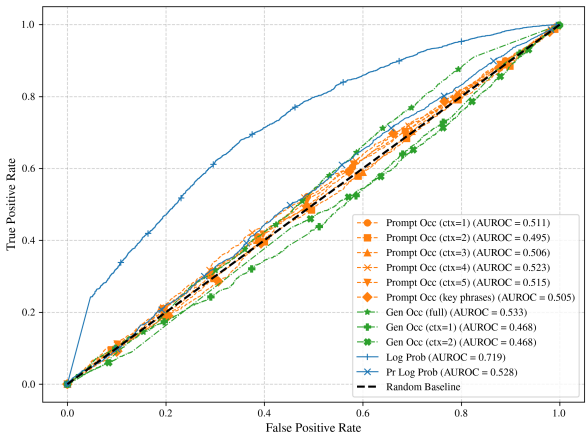
(b) TriviaQA (n-gram Model)



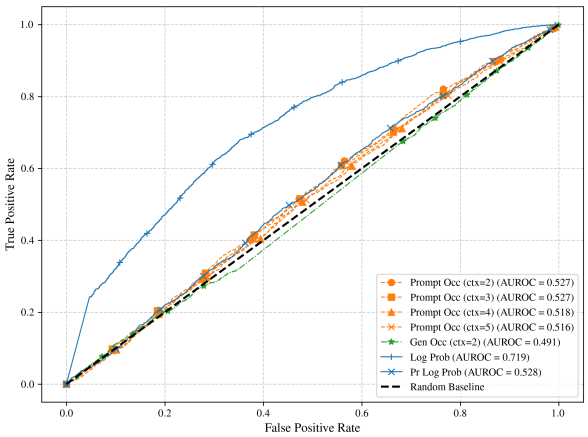
(c) NQ-Open (Raw Frequency)



(d) NQ-Open (n-gram Model)



(e) CoQA (Raw Frequency)

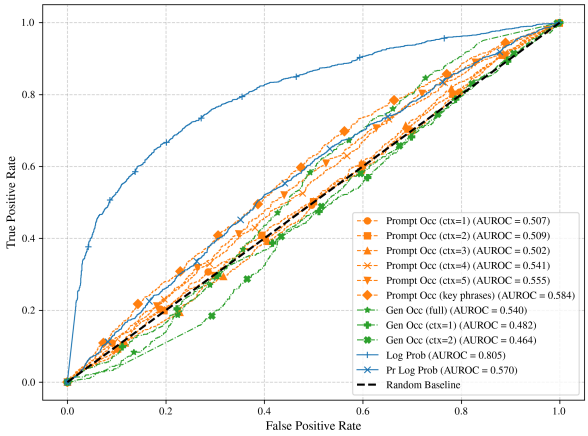


(f) CoQA (n-gram Model)

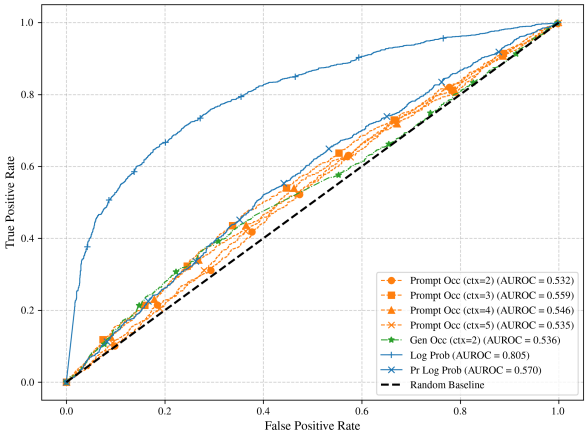
Figure 6: AUROC curves comparing log-probabilities and occurrence-based features across datasets, with stopwords filtered. Model: RedPajama-INCITE-7B; metric: ROUGE-L

Feature Type	Dataset	Model	Tree Depth	Train Acc (Log)	Train Acc (Full)	Test Acc (Log)	Test Acc (Full)
		Threshold	–	*0.728 ± 0.003*	–	*0.731 ± 0.012*	–
			3	0.748 ± 0.004	**0.752** ± 0.003	0.744 ± 0.015	**0.750** ± 0.012
			5	0.753 ± 0.004	**0.761** ± 0.004	0.740 ± 0.013	**0.748** ± 0.007
			7	0.764 ± 0.002	**0.782** ± 0.004	0.734 ± 0.015	**0.742** ± 0.009
	TriviaQA	Tree	9	0.777 ± 0.005	**0.808** ± 0.007	**0.733** ± 0.012	0.727 ± 0.012
			10	0.786 ± 0.007	**0.825** ± 0.008	**0.730** ± 0.010	0.725 ± 0.015
			20	0.892 ± 0.020	**0.973** ± 0.014	**0.699** ± 0.002	0.686 ± 0.019
		NN	–	0.746 ± 0.003	**0.752** ± 0.003	0.745 ± 0.013	**0.752** ± 0.012
		Threshold	–	*0.536 ± 0.013*	–	*0.509 ± 0.045*	–
			3	0.566 ± 0.008	**0.682** ± 0.015	0.501 ± 0.045	**0.610** ± 0.037
			5	0.603 ± 0.017	**0.763** ± 0.014	0.521 ± 0.026	**0.648** ± 0.019
			7	0.651 ± 0.014	**0.841** ± 0.009	0.509 ± 0.035	**0.637** ± 0.027
Raw Freq	NQ-Open	Tree	9	0.688 ± 0.017	**0.920** ± 0.010	0.487 ± 0.045	**0.631** ± 0.031
			10	0.710 ± 0.019	**0.948** ± 0.008	0.504 ± 0.044	**0.619** ± 0.035
			20	0.883 ± 0.045	**1.000** ± 0.000	0.501 ± 0.048	**0.640** ± 0.045
		NN	–	0.539 ± 0.009	**0.601** ± 0.014	0.518 ± 0.022	**0.552** ± 0.055
		Threshold	–	*0.657 ± 0.003*	–	*0.654 ± 0.014*	–
			3	0.662 ± 0.003	**0.674** ± 0.002	0.649 ± 0.014	**0.669** ± 0.009
			5	0.668 ± 0.003	**0.686** ± 0.007	0.645 ± 0.011	**0.656** ± 0.014
			7	0.680 ± 0.004	**0.710** ± 0.004	0.643 ± 0.012	0.643 ± 0.006
	CoQA	Tree	9	0.693 ± 0.006	**0.747** ± 0.005	**0.639** ± 0.009	0.634 ± 0.011
			10	0.701 ± 0.009	**0.768** ± 0.007	**0.640** ± 0.013	0.624 ± 0.016
			20	0.821 ± 0.023	**0.966** ± 0.011	**0.612** ± 0.013	0.582 ± 0.011
		NN	–	0.658 ± 0.003	**0.660** ± 0.003	0.653 ± 0.016	**0.656** ± 0.015
		Threshold	–	*0.728 ± 0.003*	–	*0.731 ± 0.012*	–
			3	0.748 ± 0.004	**0.751** ± 0.003	0.745 ± 0.015	**0.748** ± 0.012
			5	0.754 ± 0.004	**0.764** ± 0.004	0.740 ± 0.013	**0.747** ± 0.009
			7	0.764 ± 0.003	**0.786** ± 0.004	0.734 ± 0.015	**0.744** ± 0.017
	TriviaQA	Tree	9	0.777 ± 0.005	**0.811** ± 0.008	0.733 ± 0.012	**0.734** ± 0.015
			10	0.786 ± 0.007	**0.828** ± 0.007	**0.730** ± 0.010	0.723 ± 0.011
			20	0.892 ± 0.020	**0.970** ± 0.010	**0.699** ± 0.002	0.685 ± 0.005
		NN	–	0.746 ± 0.003	**0.749** ± 0.004	0.745 ± 0.013	**0.746** ± 0.014
		Threshold	–	*0.536 ± 0.013*	–	*0.509 ± 0.045*	–
			3	0.566 ± 0.008	**0.665** ± 0.012	0.502 ± 0.045	**0.590** ± 0.016
			5	0.603 ± 0.017	**0.704** ± 0.017	0.521 ± 0.026	**0.605** ± 0.030
			7	0.651 ± 0.014	**0.775** ± 0.018	0.509 ± 0.035	**0.606** ± 0.023
N-gram	NQ-Open	Tree	9	0.688 ± 0.017	**0.841** ± 0.019	0.487 ± 0.045	**0.578** ± 0.023
			10	0.710 ± 0.019	**0.869** ± 0.023	0.505 ± 0.044	**0.587** ± 0.022
			20	0.883 ± 0.045	**0.990** ± 0.019	0.502 ± 0.048	**0.570** ± 0.033
		NN	–	0.539 ± 0.009	**0.612** ± 0.014	0.518 ± 0.022	**0.581** ± 0.028
		Threshold	–	*0.657 ± 0.003*	–	*0.654 ± 0.014*	–
			3	0.662 ± 0.003	**0.664** ± 0.003	**0.649** ± 0.014	0.647 ± 0.011
			5	0.668 ± 0.003	**0.678** ± 0.006	**0.645** ± 0.011	0.635 ± 0.012
			7	0.680 ± 0.004	**0.702** ± 0.007	**0.643** ± 0.012	0.642 ± 0.016
	CoQA	Tree	9	0.693 ± 0.006	**0.740** ± 0.007	**0.639** ± 0.009	0.628 ± 0.020
			10	0.701 ± 0.009	**0.763** ± 0.013	**0.640** ± 0.013	0.625 ± 0.020
			20	0.821 ± 0.023	**0.958** ± 0.025	**0.612** ± 0.013	0.581 ± 0.017
		NN	–	0.658 ± 0.003	**0.663** ± 0.005	0.653 ± 0.016	**0.661** ± 0.017

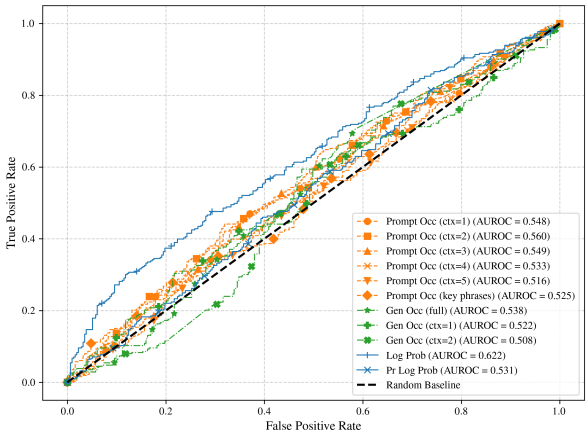
Table 7: Accuracy across tree depths, datasets, and feature types, with stopwords filtered from occurrence features. Model: RedPajama-INCITE-7B; metric: ROUGE-L.



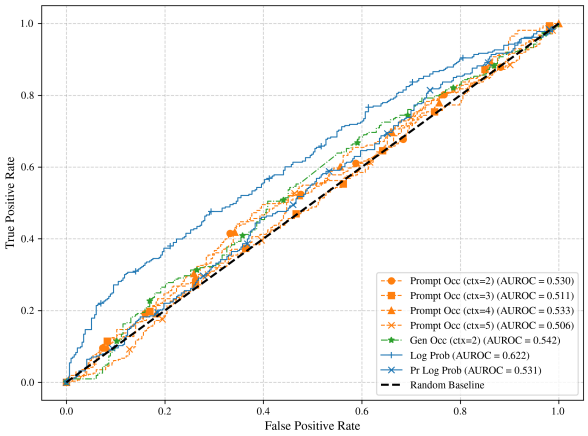
(a) TriviaQA (Raw Frequency)



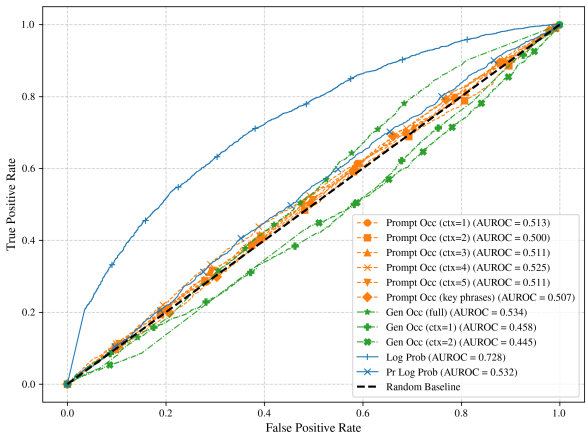
(b) TriviaQA (n-gram Model)



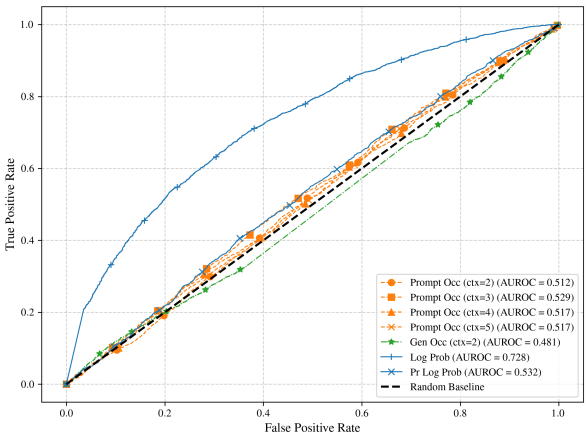
(c) NQ-Open (Raw Frequency)



(d) NQ-Open (n-gram Model)



(e) CoQA (Raw Frequency)



(f) CoQA (n-gram Model)

Figure 7: AUROC curves comparing log-probabilities and occurrence-based features across datasets. Model: RedPajama-INCITE-3B; metric: RougeL

Feature Type	Dataset	Model	Tree Depth	Train Acc (Log)	Train Acc (Full)	Test Acc (Log)	Test Acc (Full)
Raw Freq	TriviaQA	*Threshold*	–	*0.731 ± 0.004*	–	*0.724 ± 0.020*	–
		Tree	3	0.738 ± 0.003	**0.746** ± 0.004	0.720 ± 0.016	**0.749** ± 0.008
			5	0.747 ± 0.004	**0.760** ± 0.005	0.717 ± 0.021	**0.736** ± 0.013
			7	0.760 ± 0.004	**0.788** ± 0.012	0.716 ± 0.015	**0.732** ± 0.017
			9	0.780 ± 0.004	**0.829** ± 0.011	0.705 ± 0.019	**0.715** ± 0.015
			10	0.792 ± 0.004	**0.855** ± 0.011	0.692 ± 0.021	**0.709** ± 0.013
			20	0.923 ± 0.011	**0.992** ± 0.004	0.664 ± 0.017	**0.678** ± 0.027
		NN	–	0.735 ± 0.004	**0.738** ± 0.005	**0.730** ± 0.012	0.721 ± 0.013
	NQ-Open	*Threshold*	–	*0.580 ± 0.004*	–	*0.585 ± 0.019*	–
		Tree	3	0.605 ± 0.011	**0.628** ± 0.014	**0.602** ± 0.032	0.551 ± 0.037
			5	0.622 ± 0.009	**0.695** ± 0.004	**0.586** ± 0.021	0.559 ± 0.037
			7	0.643 ± 0.016	**0.770** ± 0.018	**0.576** ± 0.027	0.548 ± 0.048
			9	0.668 ± 0.026	**0.835** ± 0.024	**0.585** ± 0.021	0.548 ± 0.044
			10	0.680 ± 0.026	**0.868** ± 0.034	**0.591** ± 0.033	0.541 ± 0.043
			20	0.793 ± 0.048	**1.000** ± 0.000	**0.548** ± 0.039	0.544 ± 0.042
		NN	–	0.586 ± 0.008	**0.590** ± 0.018	**0.590** ± 0.018	0.584 ± 0.027
	CoQA	*Threshold*	–	*0.664 ± 0.003*	–	*0.665 ± 0.008*	–
		Tree	3	0.668 ± 0.002	**0.668** ± 0.007	0.661 ± 0.009	**0.664** ± 0.011
			5	0.674 ± 0.001	**0.678** ± 0.004	0.664 ± 0.004	**0.667** ± 0.017
			7	0.686 ± 0.003	**0.700** ± 0.008	**0.663** ± 0.006	0.651 ± 0.017
			9	0.701 ± 0.004	**0.735** ± 0.013	**0.655** ± 0.006	0.639 ± 0.012
			10	0.712 ± 0.004	**0.756** ± 0.016	**0.653** ± 0.008	0.633 ± 0.015
			20	0.827 ± 0.015	**0.952** ± 0.013	**0.627** ± 0.013	0.588 ± 0.012
		NN	–	0.666 ± 0.003	**0.667** ± 0.004	**0.666** ± 0.010	0.664 ± 0.010
N-gram	TriviaQA	*Threshold*	–	*0.731 ± 0.004*	–	*0.724 ± 0.020*	–
		Tree	3	0.738 ± 0.003	**0.741** ± 0.005	0.720 ± 0.016	**0.724** ± 0.011
			5	0.747 ± 0.004	**0.763** ± 0.004	**0.717** ± 0.021	0.712 ± 0.018
			7	0.760 ± 0.004	**0.793** ± 0.006	**0.716** ± 0.015	0.701 ± 0.011
			9	0.780 ± 0.004	**0.836** ± 0.009	**0.705** ± 0.019	0.686 ± 0.015
			10	0.792 ± 0.004	**0.862** ± 0.007	**0.692** ± 0.021	0.683 ± 0.010
			20	0.923 ± 0.011	**0.993** ± 0.005	**0.664** ± 0.017	0.651 ± 0.014
		NN	–	0.735 ± 0.004	**0.737** ± 0.003	**0.730** ± 0.012	0.728 ± 0.012
	NQ-Open	*Threshold*	–	*0.580 ± 0.004*	–	*0.585 ± 0.019*	–
		Tree	3	0.605 ± 0.011	**0.620** ± 0.005	**0.602** ± 0.032	0.528 ± 0.030
			5	0.622 ± 0.009	**0.690** ± 0.015	0.586 ± 0.021	**0.609** ± 0.046
			7	0.643 ± 0.016	**0.748** ± 0.019	**0.576** ± 0.027	0.566 ± 0.036
			9	0.668 ± 0.026	**0.803** ± 0.026	**0.585** ± 0.021	0.583 ± 0.053
			10	0.680 ± 0.026	**0.833** ± 0.028	**0.591** ± 0.033	0.583 ± 0.041
			20	0.793 ± 0.048	**0.986** ± 0.016	0.548 ± 0.039	**0.555** ± 0.029
		NN	–	0.586 ± 0.008	**0.601** ± 0.012	**0.590** ± 0.018	0.576 ± 0.034
	CoQA	*Threshold*	–	*0.664 ± 0.003*	–	*0.665 ± 0.008*	–
		Tree	3	0.668 ± 0.002	**0.671** ± 0.006	0.661 ± 0.009	**0.666** ± 0.012
			5	0.674 ± 0.001	**0.685** ± 0.005	0.664 ± 0.004	**0.673** ± 0.010
			7	0.686 ± 0.003	**0.711** ± 0.006	**0.663** ± 0.006	0.653 ± 0.006
			9	0.701 ± 0.004	**0.750** ± 0.010	**0.655** ± 0.006	0.642 ± 0.016
			10	0.712 ± 0.004	**0.774** ± 0.012	**0.653** ± 0.008	0.630 ± 0.020
			20	0.827 ± 0.015	**0.962** ± 0.018	**0.627** ± 0.013	0.598 ± 0.005
		NN	–	0.666 ± 0.003	**0.672** ± 0.005	0.666 ± 0.010	**0.673** ± 0.008

Table 8: Accuracy across tree depths, datasets, and feature types. Bold indicates higher accuracy between log-only and full features. Model: RedPajama-INCITE-3B; metric: ROUGE-L.

Dataset	N-gram	% Zeros	Examples with zero occurrence in training data
TriviaQA	1	0.00	–
	2	0.03	mental originate | aston ban | ovsk refer
	3	3.33	keyboard the % | iva' released | 56 balls?
	4	14.97	Jersey Boys"" | ove are types of | which famous product?
	5	31.99	what is classified using the | Who (at 2008) | "In Texas, what
	key phrases	41.30	there will be blood, film | actor, second Dr Who | December 2, 1805, battle
NQ-Open	1	0.00	–
	2	0.06	slew win | today gest | gium invade
	3	4.79	when does bill | s eg wells | plays dorian
	4	21.09	build tables that identify | ondon finished being built | sings oh what a
	5	44.44	boundary was the mexico | who played zoe h | does the paraguay
	key phrases	55.48	completion of tower of london | ubuntu project founder | macos operating system
CoQA	1	0.00	–
	2	0.02	TribeII | Collect Doyle | compensatory innocence
	3	1.56	golden dog Kay | upon, Johann | Rex ran home
	4	9.40	of prosecutors argument that | away shooting; for | to learn French signs
	5	24.80	asked king to send him | man on that houseboat | Charity found Arthur would
	key phrases	39.37	sheriff norman chaffins | nokobee woods | new york fashion internships

Table 9: Percentage of zero-occurrence n-grams and key phrases in the training corpus