The Rarity Blind Spot: A Framework for Evaluating Statistical Reasoning in LLMs

Anonymous ACL submission

Abstract

Effective decision-making often relies on identifying what makes each candidate distinctive. While existing benchmarks for LLMs emphasize retrieving or summarizing information relevant to a given query, they do not evaluate a model's ability to identify globally distinctive features across a set of documents. We introduce Distinctive Feature Mining (DFM), a new task that challenges models to analyze a small-to-medium collection (10-40 documents) and surface features that are rare in the global context (e.g., appearing in less than 10% of documents). This setting mirrors real-world scenarios such as candidate selection or product differentiation, where statistical reasoning, not retrieval, is key. To enable systematic evaluation of this capability, we present DIFBENCH, a configurable benchmark creation framework with controllable parameters such as document set size and distinctiveness thresholds.

Using DIFBENCH, we perform a large-scale assessment of distinctive feature mining across ten state-of-the-art LLMs. Our findings reveal a significant performance gap between general-purpose and reasoning-enhanced models. All models, however, substantially degrade as the task complexity and document count increase. We also find that a common failure mode is misidentifying frequent features as distinctive. These insights reveal core limitations in contemporary LLMs' abilities to perform fine-grained, statistical reasoning and rarity detection.

1 Introduction

When making decisions from large candidate pools–whether selecting products, evaluating applicants, or analyzing documents–humans naturally seek to understand what makes each candidate distinctive. This cognitive process of identifying uncommon or unique traits is central to effective decision-making.

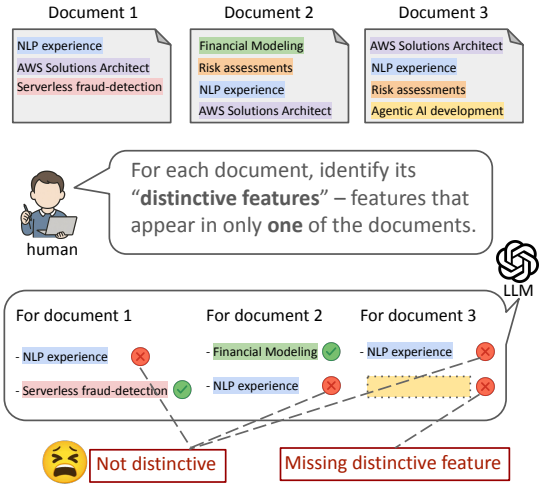


Figure 1: Example of Distinctive Feature Mining (DFM). Given a set of documents, the model needs to identify globally rare features. Here, the model incorrectly identifies *"NLP experience"* as distinctive, when it is shared by all documents. In contrast, it misses the truly rare feature *"Agentic AI development"*.

As LLMs are increasingly deployed in recommendation and decision support systems across domains such as hiring (An et al., 2024; Iso et al., 2025) and travel planning (Xie et al., 2024), their ability to mimic this core human capability becomes critical.

Our investigation reveals a fundamental limitation: even state-of-the-art reasoning models fail to recognize rarity when analyzing a set of documents. For instance, when analyzing technical resumes, a model might mistakenly identify *"NLP experience"* as distinctive when it is shared by multiple documents, and yet miss genuinely rare skills like *"Agentic AI development"* (see Figure 1). This behavior is akin to the psychological phenomenon of *base rate neglect* (Tversky and Kahneman, 1974; Grether and Plott, 2012), where statistical frequency is ignored in favor of more salient but less informative cues. This can lead to systematically suboptimal recommendations and decisions.

LLM benchmarks have primarily focused on query-driven tasks, such as sparse information retrieval (e.g., the needle-in-a-haystack test (Kamradt, 2023)) or multi-document and long-context reasoning (Karpinska et al., 2024; Xu et al., 2024; Kuratov et al., 2024; Levy et al., 2024; Bai et al., 2024a; Zhang et al., 2024a; Hsieh et al., 2024; Bai et al., 2024b; Yen et al., 2025; Maekawa et al., 2025). These benchmarks assess a model's ability to find or aggregate relevant information, often in response to an explicit query. However, they do not test whether a model can derive global statistical insights across a collection, in particular those involving feature rarity.

To fill this gap, we introduce **Distinctive Feature Mining (DFM)**, a new task that requires identifying globally rare attributes (appearing in $\leq \theta\%$ of documents) within document collections. Unlike traditional retrieval or summarization, DFM requires *statistical reasoning* over a population, not just extracting salient information from individual documents. We focus on collections of 10–40 documents, a realistic scale for decisions like candidate screening or product comparison. This scale is large enough to require aggregate reasoning and base-rate estimation, yet small enough to demand holistic comprehension and accurate attribution.

We operationalize this through **DiFBench**, a configurable benchmark creation framework that precisely governs feature distributions. For example, it ensures *"blockchain development"* appears in 2 out of 40 resumes (5%) while *"project management"* appears in 25 (62.5%). This enables systematic evaluation across document scales and domains, with controllable parameters including document count (10-40), feature density, and distinctiveness thresholds (2.5%-20%).

Our evaluation over 10 state-of-the-art LLMs reveals three key findings: (1) non-reasoning models achieve F1 < 30%, revealing limitations in multi-document reasoning; (2) even advanced models (o3, Gemini-2.5-Flash) degrade from F1 > 85% on 10 documents to F1 < 60% on 40 documents; and (3) 75.9% of errors involve misclassifying common features as distinctive. This precision drop mirrors base rate neglect in human cognition. We mitigate this via explicit verification prompting, achieving a 65% relative F1 gain while maintaining recall.

The main contributions of this work include:

(1) We introduce **DFM** task and **DiFBench** benchmark creation framework, to enable systematic evaluation of collection-level statistical reasoning across domains (resumes, news summaries), document scales (10-40), and distinctiveness thresholds (2.5%-20%).

(2) We conduct the first large-scale study revealing that even leading LLMs degrades significantly with scale, with 75.9% of errors resulting from misidentifying frequent features as distinctive. This provides computational evidence of base rate neglect in LLM reasoning.

(3) We demonstrate that explicit verification prompting leads to a 65% relative improvement in the F1 score, offering a practical mitigation while highlighting persistent limitations in multi-document comparative reasoning.

We will release the datasets and the evaluation framework upon acceptance of this paper.

2 Related Work

Complex and Quantitative Reasoning in LLMs Recent benchmarks increasingly test multi-document reasoning, but their primary focus remains on aggregating query-relevant content or retrieving salient passages (Levy et al., 2024; Bai et al., 2024a; Zhang et al., 2024a; Hsieh et al., 2024; Bai et al., 2024b; Yen et al., 2025; Maekawa et al., 2025). In contrast, DFM shifts the focus to corpus-level statistical reasoning, requiring the identification of globally rare features. This requires reliable counting, base rate estimation, and population-level comparison. These are all areas where LLMs remain weak (Maekawa et al., 2025). Our findings reinforce this, showing that models often miscount feature frequencies and overestimate the distinctiveness of common traits. These limitations highlight statistical reasoning across documents as an underexplored and unresolved challenge.

Multi-document Summarization Multi-document summarization typically aims to synthesize common themes or provide a unified overview of content across documents (Li et al., 2012; Laban et al., 2024; Belem et al., 2025). A few recent efforts (Huang et al., 2024; Zhang et al., 2024b) have explored diversity-aware summarization, but they focus on maximizing coverage of perspectives rather than surfacing rare or distinctive features. DFM complements this line of work by targeting corpus-level rarity rather than within-document salience or inter-document consensus.

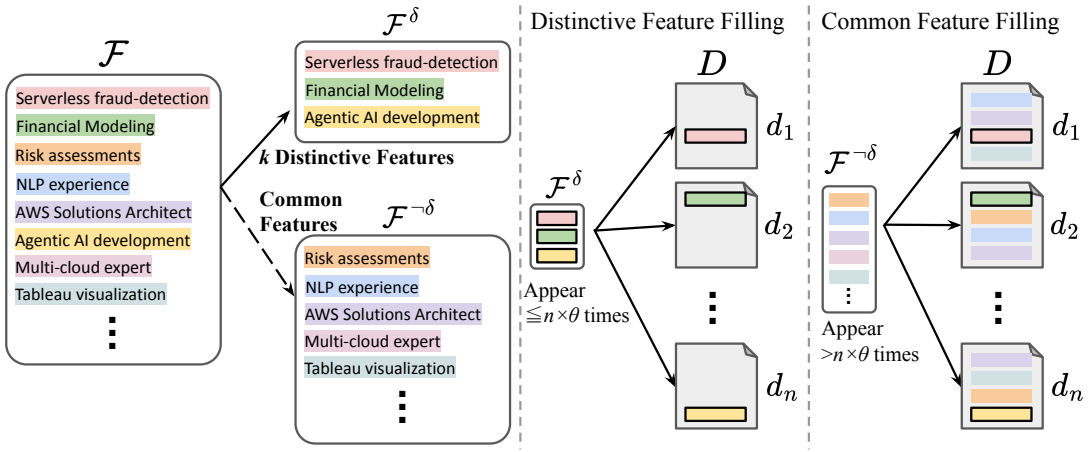


Figure 2: Overview of DIFBENCH. To obtain distinctive features, $\mathcal{F}^{\delta}$, we first randomly select k features from the feature set $\mathcal{F}$. The remaining features are treated as common features, $\mathcal{F}^{\neg\delta}$. Distinctive features $\mathcal{F}^{\delta}$ are distributed across documents while ensuring that each feature appears less than or equal to $\theta\%$ of the documents. Common features $\mathcal{F}^{\neg\delta}$ are then distributed across documents, ensuring that each feature appears in more than $\theta\%$ of the n documents.

Comparative Summarization and Pairwise Analysis Prior work on comparative summarization has explored pairwise document contrast and entity differentiation (Iso et al., 2022; Gunel et al., 2023, 2024; Yan et al., 2024). These methods effectively highlight differences between two entities but do not scale to collections with many candidates. Crucially, they also lack a statistical frame for identifying what is rare relative to a population. DFM extends these efforts to multi-way comparisons, allowing models to reason over the distinctiveness of features in the context of an entire set—a key requirement in realistic decision-making scenarios such as hiring or product recommendation.

3 Distinctive Feature Mining and Benchmark Creation Framework

We first introduce the task of Distinctive Feature Mining (DFM), present the design principles of DIFBENCH, a general benchmark creation framework designed to systematically evaluate models on this task. Then, we explain the details of the benchmark creation framework. Finally, we describe how DIFBENCH is implemented to create benchmark datasets.

3.1 Task Definition

In this study, we simplify each document into a set of features. This can be realized by feature extraction methods (Clavié and Soulié, 2023) in common use cases such as resume screening and product comparison. Formally, a document set is denoted as $D = \{d_1, d_2, \ldots, d_n\}$, where each document d_i consists of a set of up to h features, $F_i = \{f_1^i, f_2^i, \ldots\}$. Let $\mathcal{F} = \bigcup_{i=1}^{n} F_i$ denotes the set of all features across D. The task of Distinctive Feature Mining (DFM) is to identify, for each document d_i, a subset of features $F_i^{\delta} \subseteq F_i$ that are distinctive. A feature is considered *distinctive* if it appears in at most $\theta\%$ of documents, where θ is a user-defined threshold.

3.2 Design Principles

We introduce DIFBENCH, a benchmark creation framework specifically designed to evaluate model performance on the DFM task. Figure 2 illustrates the overall process. Given a feature set $\mathcal{F}$, the framework partitions it into distinctive and common subsets, then distributes these features across documents D in a controlled manner to enable systematic evaluation. DIFBENCH is guided by three core design principles:

(1) **Distinctive features in comparable candidates**: Documents must be comparable; that is, all documents belong to the same domain and share the same structure. They differ in select features that make them distinctive.

(2) **Flexible number of candidates and distinctive features**: Variable numbers of candidate documents and distinctive features must be supported to enable evaluation across scale and distinctiveness thresholds.

(3) **Systematic evaluation**: The framework must

enable controlled experiments and facilitate precise measurement of model ability to detect globally rare features and reason over aggregate statistics.

These principles enable a comprehensive testbed for studying corpus-level statistical reasoning. It allows researchers to probe models' capacity to (1) extract features across documents, (2) count their frequencies, and (3) identify what is statistically distinctive in a given population.

3.3 Benchmark Creation Framework

To realize these design principles in practice, DIFBENCH takes as input a set of features $\mathcal{F}$ and programmatically constructs a document set D by distributing these features based on configurable parameters. These key parameters are:

(1) **Number of documents** (n): Controls the scale of the dataset, allowing us to test how model performance varies with small to large document collections. Increasing n raises the complexity of DFM as models must consider more candidates and interactions.

(2) **Number of distinctive features** (k): Specifies how many features are truly distinctive across the document set. By varying k, we can simulate settings where distinctive traits are sparse or abundant, which affects the difficulty of mining such features.

(3) **Distinctiveness threshold** (θ): Defines the maximum proportion of documents a feature can appear in to be considered distinctive. This parameter enables us to influence feature rarity and overlap across documents.

Together, these provide fine-grained control over the complexity, sparsity, and overlap within the benchmark, enabling systematic and reproducible evaluation of statistical reasoning capabilities.

Document Set Construction The benchmark creation process begins by distributing distinctive features across a subset of documents, followed by populating the remaining feature slots with common features. To this end, we first randomly select k features from the set of features, $\mathcal{F}$, to serve as distinctive features, $\mathcal{F}^\delta$. Each distinctive feature is assigned a target document frequency, randomly sampled from the range $[1, n \times \theta]$, ensuring these features appear in only a small portion of the n documents. These distinctive features are then distributed across the documents to match their assigned frequencies. The remaining features $F^{\neg\delta} = \mathcal{F} \setminus \mathcal{F}^\delta$ are treated as common features. Each is assigned a higher document frequency, sampled from the range $[n \times \theta + 1, n]$, and distributed across documents in the same way. During assignment, we enforce a constraint that each document can contain at most h features. If a feature cannot be assigned without violating this rule, its assignment is skipped. This ensures that distinctive features remain relatively rare within the document set, while common features are broadly shared, thus preserving the intended distinction between the two categories.

3.4 Benchmark Implementation

While DIFBENCH is designed to accept any set of features, our implementation focuses on synthesizing features grounded in real-world source documents. Rather than relying on exact feature extraction, we opt for feature synthesis to support systematic and controlled evaluations. This approach ensures the generated features remain realistic and representative of the original documents while allowing us to precisely control task complexity.

Data Domains We use two different domains: resumes and news summaries—both well-suited for comparative analysis. For resumes, we source job posts from `mycareerfuture.sg`,[1] selecting the 10 longest descriptions from each of five major job categories based on US Department of Labor statistics,[2]. These include computer & math, life physical & social science, legal, architecture & engineering, and healthcare occupations. For news summaries, we utilized news articles from (Huang et al., 2024), which cover five distinct topics. Each topic has 10 news articles. In summary, we have 100 source documents in total, with 50 resumes and 50 news summaries.

Feature Set Generation For each source document, we synthesize a set of features that are both relevant to its content and representative of its context. To guide feature creation, we use domain-specific structural templates. For resumes, these include categories such as Experience, Technical Skills, Soft Skills, Projects, Certifications,

[1]We downloaded the dataset from https://github.com/WING-NUS/JD2Skills-BERT-XMLC

[2]Labor Force Statistics from the Current Population Survey: https://www.bls.gov/cps/cpsaat11.htm

Domain	Section	Synthesized Feature
Resume	Experience	Architected multi-cloud application frameworks aligning with banking industry compliance.
	Technical Skills	Proficient in .NET Framework and .NET Core architectures
	Soft Skills	Facilitated transparent communication across technical and non-technical audiences
	Projects	Built serverless fraud-detection prototype leveraging AWS Lambda streams
	Certifications	Achieved AWS Solutions Architect – Professional certification
	Awards and Recognition	Earned Global Cloud Excellence award for innovative platform design
News Summary	fuel requirements	Inadequate ethanol content could trigger knock sensors and limp-home modes, ruining track sessions.
	vehicle performance	Carbon-fiber rim option trims 32 pounds of unsprung mass, quickening initial acceleration.
	historical context	Factory 1,000-hp rating revives 1960s "horsepower wars" in a final escalation.
	optional features	$10,000 sunroof pricing intentionally discourages extra roof weight.
	NHRA regulations	Street-legal Demons may drive NHRA to revisit Advanced ET class definitions.
	production details	Compressed 2023 build window heightens risk of missed quotas before Brampton plant closure.
	branding and marketing	Devilish $96,666 base price turns MSRP into instant viral talking point.

Table 1: Examples of synthesized features in the resume and news summary dataset.

and Awards. For news summaries, we adopt 7–9 subtopics from the original dataset (e.g., `fuel requirements` under the motor trend topic) as section headers.

For each section, we prompt an LLM to generate a pool of 20 thematically relevant candidate features, using the seed document and section title as context. To encourage diversity across sections, we also supply the model with previously generated features from other sections of the same document. This helps ensure that each section's features are both semantically relevant and distinct. We employ o3 (OpenAI, 2025) for a feature generation. Table 1 illustrates several examples of synthesized features.

4 Experimental Setup

This section outlines our methodology for evaluating the statistical reasoning capabilities of LLMs. We first describe the parameters used for generating the synthetic document collections using DIFBENCH. We then introduce the suite of LLMs evaluated in our experiments, followed by a description of our inference and evaluation setup.

Document Set Construction Parameters We set the number of documents n to 10, 20, and 40 and test with $\lfloor n/2 \rfloor$ distinctive features, to examine how LLMs handle varying levels of complexity in identifying distinctive features. We set the distinctive threshold θ to 2.5%, 5%, 10%, and 20% of the total documents (i.e., 1, 2, 4, and 8 documents respectively when $n = 40$). We set the maximum number of features per document h to $4 \times S$, where S denotes the number of sections of the document.

Models We evaluate 10 LLMs with reasoning-optimized and general-purpose capabilities. Reasoning models include both closed and open models: o3, o4-mini, Gemini-2.5-Flash, Qwen3-235B22A (Qwen3 for short). General models include GPT-4o, GPT-4o-mini, Gemini-2.5-Flash w/o think, Qwen3 w/o think, Llama-4-Maverick, and Llama-4-Scout. The model details are summarized in Appendix B. We set temperature and top-p parameters to 0.0 and 1.0, respectively, for all our experiments.

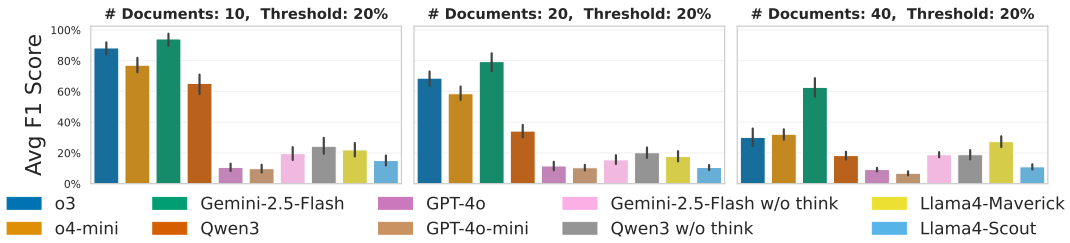
Inference Setup At inference time, each model is presented with a collection of documents generated by DIFBENCH and tasked with identifying the set of distinctive features within that collection. The model receives a single instruction prompt that asks it to return the features that appear rarely (distinctive features) for each document. Because DIFBENCH controls the construction of documents and explicitly selects which features are to be distinctive, we have access to ground-truth annotations $\mathcal{F}^{\delta}$ for each synthetic benchmark instance. This setup allows for objective evaluation of model predictions against a known gold standard. We use the same prompt for all models, see Appendix C.

Evaluation Metrics Model predictions are evaluated using exact string match against the ground-truth set $\mathcal{F}^{\delta}$ provided by DIFBENCH. Our primary evaluation metric is the F1 score, with precision and recall reported in detailed analyses.

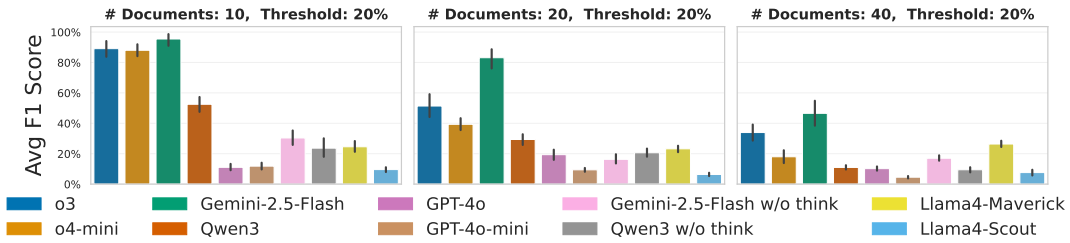
5 Results and Analysis

5.1 Main Results

Reasoning models consistently outperform their general counterparts. Table 2 shows average F1 scores on the DFM task across varying document counts (10, 20, 40) and distinctiveness thresholds (10% and 20%). Overall, reasoning models consistently outperform general-purpose



(a) Resumes



(b) News summaries

Figure 3: F1 scores with various document sizes. The error bars indicate the standard deviation across samples.

Models	Resumes	News Summaries
o3	68.95	69.81
o4-mini	61.92	58.45
Gemini-2.5-Flash	84.78	77.76
Qwen3	46.41	36.32
GPT-4o	12.55	17.12
GPT-4o-mini	8.45	7.77
Gemini-2.5-Flash w/o think	20.38	22.29
Qwen3 w/o think	24.08	18.72
Llama4-Maverick	25.89	25.34
Llama4-Scout	11.87	7.21

Table 2: Average F1 scores of the reasoning-optimized and general-purpose models on the DFM task across three document sizes, i.e., 10, 20, and 40, and two distinctive features, i.e., 10% amd 20%. The models with *w/o* suffix are general models that do not use reasoning capabilities.

models across all settings. Surprisingly, no general model achieves F1 higher than 0.3%, indicating their limitation in identifying distinctive features effectively. This is particularly evident when comparing Gemini-2.5-Flash and Qwen3 with their non-reasoning ('w/o think') variants, where reasoning-optimized versions consistently perform better. This trend holds across both domains, resumes and news summaries.

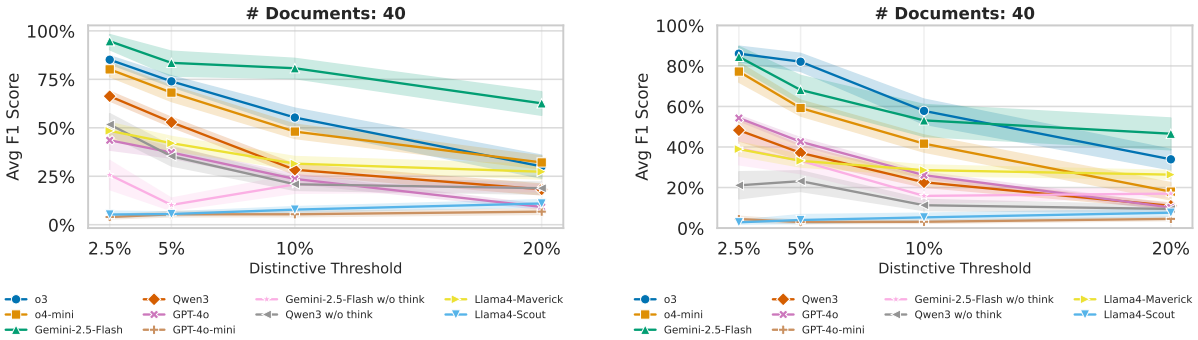
Even current reasoning models are poor statistical reasoners when the collection size increases. To investigate the impact of number of documents on DFM performance, we break down the results by document size in Figure 3, focusing on the 20% distinctive threshold. Results for 10% threshold are included in the Appendix A.1, where we observe similar trends.

Reasoning models consistently outperform general models across all document sizes, with F1 scores generally degrading as the number of documents increases. Their advantage is most evident with smaller sets (10 documents), where models like o3 and Gemini-2.5-Flash achieve F1 scores above 85%. However, performance drops sharply as the number of documents increases, with F1 scores dropping below 60% for 40 documents in most cases. This suggests that while reasoning capabilities significantly benefit DFM, current models still struggle with the multi-document comparison at larger scales.

Statistical reasoning becomes more challenging as the distinctive threshold increases. We further analyze F1 scores across varying distinctive thresholds, keeping the number of documents fixed at 40 (see Figure 4). We observe that F1 scores generally decline as the threshold increases, suggesting it becomes harder for models to isolate features that distinguish fewer documents from a larger set. At higher thresholds (e.g., 20%), all models perform poorly, with narrow gap between reasoning and general models. This implies that finer-grained DFM still remains a key challenge even for advanced reasoning models.

5.2 Analysis of DFM Performance

We conduct a deeper analysis of LLM performance on the DFM task by examining precision, recall and



(a) Resumes

(b) News summaries

Figure 4: F1 scores with 40 documents and various distinctive thresholds.

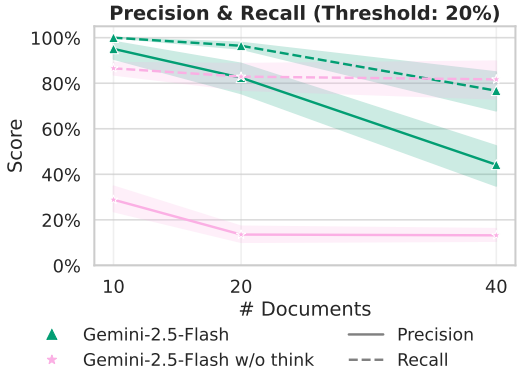


Figure 5: The precision and recall

token usage. Due to space constraints,, we present detailed results on the news summary dataset here and include additional results on the resume dataset in Appendix A.2, which follow similar trends.

While general models tend to over-predict, reasoning models are more selective in identifying truly distinctive features. Figure 5 shows the average precision and recall of Gemini-2.5-Flash and its w/o think variant across various document sizes. Gemini-2.5-Flash generally achieves higher precision than its w/o think, suggesting that reasoning models are more effective at identifying truly distinctive features. Interestingly, general models achieve relatively higher recall but poor precision (<30%), suggesting they tend to over-predict and include many irrelevant features. This reflects a lack of selectivity in general models when attempting feature mining under increasing complexity. Finally, we observe that both precision and recall drop as the document size increases even for the best-performing reasoning model, Gemini-2.5-Flash. This indicates that as the number of documents increases, models struggle to accurately count feature occurrences and identify those that are truly distinctive.

Better statistical reasoning requires more output tokens. Figure 6 shows the average number of output tokens under varying document sizes. As shown, reasoning models tend to generate more tokens in total as the number of documents increases. This indicates that models require more reasoning to identify distinctive features when the complexity of the task increases. Notably, when considered alongside the results in Figure 3, Gemini-2.5-Flash achieves a high F1 score by significantly increasing its token usage compared to other models. We also observe that Gemini-2.5-Flash w/o think generates a larger number of tokens than most of other models, despite its low precision score (see Figure 5). This suggests that the model struggles with the statistical reasoning even if it generates a large amount of tokens to identify rare information. The results on the resume dataset is included in Appendix A.3, which shows similar trends.

5.3 Error Analysis

To better understand model limitations, we analyze the errors made by the best-performing model, Gemini-2.5-Flash, in the most challenging setting (40 documents and a distinctive threshold of 20%). We categorize the errors into three main types: (1) **Non-distinctive.** Features that are mentioned in the document but are not distinctive. (2) **Contamination.** Features that are not mentioned in the document itself but occur in other documents. (3) **Typo/Abbreviation.** Features that are not mentioned in any documents, often due to typos or malformed abbreviations.

The best performing model still struggles to estimate frequencies of features. Table 3 shows

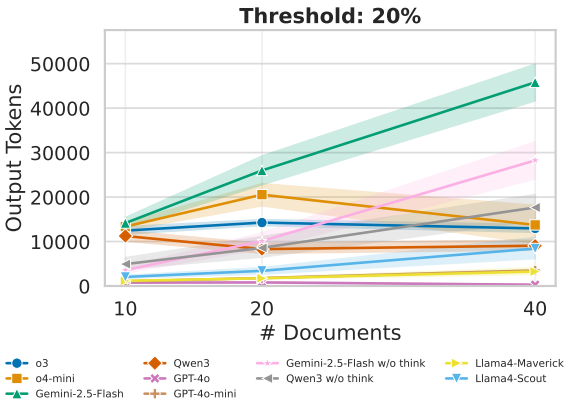


Figure 6: The average number of output tokens.

Category	Percentage (%)
Non-distinctive	75.90
Contamination	1.89
Typo/Abbreviation	0.01
Correct	22.20

Table 3: The distribution of error categories.

the distribution of these errors. The majority of errors are non-distinctive features, which indicates that the model tends to identify features that appear in the document but are not truly distinctive. This result suggests that models struggle to correctly estimate the frequencies of features if they handle many features. We also observe that the model makes contamination errors, which indicates that the model tends to identify features that are not mentioned in the document but are present in other documents. Since the recent model follows the instruction well, we observe that the model rarely makes typo/abbreviation errors.

5.4 Mitigation Strategy

Motivated by the observations that models achieve high precision but low recall in Figure 5 and the majority of errors comes from non-distinctive features in Table 3, we propose a simple post-processing strategy to mitigate the errors. The mitigation strategy is as follows: 1) for each model-generated feature, we have the model judge whether the feature is distinctive or not by comparing the feature with all documents, and 2) if the feature is distinctive, we keep it; otherwise, we discard it. This strategy is based on the assumption that if the model processes features one by one, it can better identify whether the feature is distinctive or not than processing all features at once.

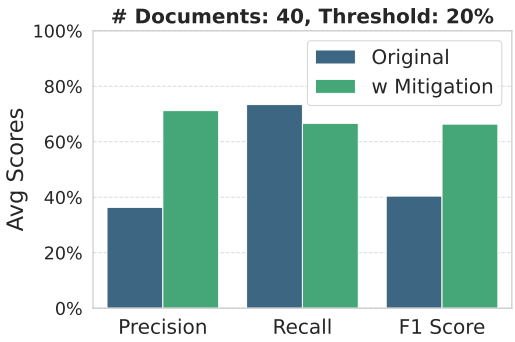


Figure 7: Effect of our mitigation strategy.

The mitigation strategy dramatically improves the precision of the DFM task but it is still far from perfect. Figure 7 shows the F1 scores of the DFM task with and without the mitigation strategy. We use Gemini-2.5-Flash as a judge model and other settings are the same as Table 3. We observe that the mitigation strategy successfully improves the precision with a slight decrease in recall, resulting in a higher F1 score which is 65% improvement over the original score. However, the mitigation strategy requires high cost and adds latency , as it needs to compare each model-generated feature with all documents to check whether the feature is distinctive or not. Also, while the mitigation strategy improves the precision, it is still around 70%, which indicates that the model still struggles to identify distinctive features correctly even if it processes features one by one. We leave the exploration of more efficient and reliable mitigation strategies for future work.

6 Conclusion

We introduced DIFBENCH, a configurable benchmark creation framework for the distinctive feature mining (DFM) task, designed to systematically evaluate the statistical reasoning capabilities of LLMs. Through extensive experiments with ten models—four reasoning and six general-purpose—we found that reasoning models consistently outperform general ones but degrade sharply as the number of documents increases. Our analysis revealed that even strong models struggle with precision and often misidentify common features as distinctive. While our mitigation strategy substantially improved precision, it also emphasized the need for more robust methods to support reliable statistical reasoning in complex multi-document settings.

Limitations

While this work systematically evaluates how LLMs identify distinctive single feature, a logical next step is to explore combinational distinctiveness. Examining unique combinations of features would more accurately reflect the complexity of real-world scenarios.

Since the purpose of our work is to assess the capabilities of LLMs in statistical reasoning, we simply evaluate every feature in the document set without considering their weight or importance. The evaluation framework does not account for the weight of features, which can be a significant factor in determining their importance. In real-world scenarios, some features may carry more weight than others, influencing their relevance and distinctiveness. For instance, in a resume analysis context, certain skills or experiences may be more critical than others. Future work could explore methods to incorporate feature weighting into the evaluation process, allowing for a more nuanced assessment of the distinctive feature mining.

Additionally, our current evaluation relies on an exact string match, which simplifies the task. A natural next step for future work is to increase the complexity by incorporating paraphrased features. This would require models to identify semantically equivalent but textually different features, making the benchmark more challenging and aligned with real-world complexities.

Ethical considerations

We acknowledge the ethical implications of Distinctive Feature Mining (DFM), particularly in high-stakes domains like hiring. While our work is a technical exploration of statistical reasoning, its application requires careful foresight. Key concerns include:

- **Bias Amplification and Proxy Discrimination:** DFM identifies statistically rare features without semantic understanding. This risks flagging features that are proxies for protected attributes (e.g., race, gender, age), potentially amplifying societal biases in downstream applications.

- **Novelty vs. Competency:** The task's focus on rarity may lead to prioritizing novel features over core competencies. This could undervalue well-qualified candidates with standard skill sets in favor of those with unique but less relevant attributes.

- **Reductionism and Dehumanization:** A feature-centric view is inherently reductionist, simplifying complex entities like candidates into a list of keywords. This risks a dehumanizing evaluation process that overlooks holistic qualities like critical thinking or creativity.

Future work must address these risks, for instance by developing fairness-aware frameworks that can distinguish between meaningful and potentially discriminatory features. We release our benchmark to encourage community research into both the capabilities and societal risks of this technology.

While we used AI assitants such as ChatGPT and Copilot to assist in coding and revising this paper, we carefully reviewed and edited all content to ensure it meets our standards and aligns with our research goals.

References

Haozhe An, Christabel Acquaye, Colin Wang, Zongxia Li, and Rachel Rudinger. 2024. Do large language models discriminate in hiring decisions on the basis of race, ethnicity, and gender? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 386–397, Bangkok, Thailand. Association for Computational Linguistics.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024a. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024b. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*.

Catarina G Belem, Pouya Pezeskhpour, Hayate Iso, Seiji Maekawa, Nikita Bhutani, and Estevam Hruschka. 2025. From single to multi: How llms hallucinate in multi-document summarization. In *Findings of the Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL)*.

Benjamin Clavié and Guillaume Soulié. 2023. Large language models as batteries-included zero-shot esco skills matchers. *arXiv preprint arXiv:2307.03539*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

David M. Grether and Charles R. Plott. 2012. Seeing is believing: Priors, trust, and base rate neglect. *Organizational Behavior and Human Decision Processes*, 119(1):1–14.

Beliz Gunel, Sandeep Tata, and Marc Najork. 2023. Strum: Extractive aspect-based contrastive summarization. In *Companion Proceedings of the ACM Web Conference 2023*, page 28–31.

Beliz Gunel, James B Wendt, Jing Xie, Yichao Zhou, Nguyen Vo, Zachary Fisher, and Sandeep Tata. 2024. Strum-llm: Attributed and structured contrastive summarization. *arXiv preprint arXiv:2403.19710*.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*.

Kung-Hsiang Huang, Philippe Laban, Alexander Fabbri, Prafulla Kumar Choubey, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. 2024. Embrace divergence for richer insights: A multi-document summarization benchmark and a case study on summarizing diverse information from news articles. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 570–593, Mexico City, Mexico. Association for Computational Linguistics.

Hayate Iso, Pouya Pezeshkpour, Nikita Bhutani, and Estevam Hruschka. 2025. Evaluating bias in LLMs for job-resume matching: Gender, race, and education. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 672–683, Albuquerque, New Mexico. Association for Computational Linguistics.

Hayate Iso, Xiaolan Wang, Stefanos Angelidis, and Yoshihiko Suhara. 2022. Comparative Opinion Summarization via Collaborative Decoding. In *Findings of the Association for Computational Linguistics (ACL)*.

Gregory Kamradt. 2023. Needle in a haystack - pressure testing llms. *Github repository*.

Marzena Karpinska, Katherine Thai, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. One thousand and one pairs: A "novel" challenge for long-context language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17048–17085, Miami, Florida, USA. Association for Computational Linguistics.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Igorevich Sorokin, Artyom Sorokin, and Mikhail Burtsev. 2024. BABILong: Testing the limits of LLMs with long context reasoning-in-a-haystack. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Philippe Laban, Alexander Fabbri, Caiming Xiong, and Chien-Sheng Wu. 2024. Summary of a haystack: A challenge to long-context LLMs and RAG systems. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9885–9903, Miami, Florida, USA. Association for Computational Linguistics.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15339–15353, Bangkok, Thailand. Association for Computational Linguistics.

Jingxuan Li, Lei Li, and Tao Li. 2012. Multi-document summarization via submodularity. *Applied Intelligence*, 37(3):420–430.

Seiji Maekawa, Hayate Iso, and Nikita Bhutani. 2025. Holistic reasoning with long-context LMs: A benchmark for database operations on massive textual data. In *The Thirteenth International Conference on Learning Representations*.

Meta. 2025. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation. *Technical report*.

OpenAI. 2024. Model release blog: Gpt-4o. *Technical report*.

OpenAI. 2025. Model release blog: Introducing openai o3 and o4-mini. *Technical report*.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Amos Tversky and Daniel Kahneman. 1974. Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157):1124–1131.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. 2024. Travelplanner: A benchmark for real-world planning with language agents. In *Forty-first International Conference on Machine Learning*.

Xiaoyue Xu, Qinyuan Ye, and Xiang Ren. 2024. Stress-testing long-context language models with lifelong icl and task haystack. *arXiv preprint arXiv:2407.16695*.

Jing Nathan Yan, Tianqi Liu, Justin Chiu, Jiaming Shen, Zhen Qin, Yue Yu, Charumathi Lakshmanan, Yair Kurzion, Alexander Rush, Jialu Liu, and Michael Bendersky. 2024. Predicting text preference via structured comparative reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10040–10060, Bangkok, Thailand. Association for Computational Linguistics.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. 2025. HELMET: How to evaluate long-context models effectively and thoroughly. In *The Thirteenth International Conference on Learning Representations*.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024a. ∞Bench: Extending long context evaluation beyond 100K tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277, Bangkok, Thailand. Association for Computational Linguistics.

Yusen Zhang, Nan Zhang, Yixin Liu, Alexander Fabbri, Junru Liu, Ryo Kamoi, Xiaoxin Lu, Caiming Xiong, Jieyu Zhao, Dragomir Radev, Kathleen McKeown, and Rui Zhang. 2024b. Fair abstractive summarization of diverse perspectives. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3404–3426, Mexico City, Mexico. Association for Computational Linguistics.

A Additional Results

In this section, we present additional results on the resume dataset to complement the main results presented in the paper.

A.1 Results with Distinctive Threshold 10%

Figure 8 shows the F1 scores with various document sizes when the distinctive threshold is set to 10%. Overall, the results follow similar trends to those with 20% threshold (Figure 3), i.e., reasoning models outperform general-purpose models especially when the document size is small. Aslo, we observe that the F1 scores are generally higher than those with 20% threshold (Figure 3), indicating that it is easier for models to identify distinctive features when the threshold is lower. This corresponds to our findings in Figure 4, where the F1 scores generally decrease as the distinctive threshold increases.

A.2 Precision and Recall on Resume Dataset

Figure 9 shows the precision and recall of Gemini-2.5-Flash and its w/o think variant on the DFM task with various document sizes when the distinctive threshold is set to 20%. We observe that the trends are similar to those in Figure 5 for the news summary dataset. While Gemini-2.5-Flash achieves higher precision than its w/o think variant, the precision declines as the document size increases.

A.3 Token Usage on Resume Dataset

Figure 10 shows the average number of output tokens with various document sizes when the distinctive threshold is set to 20%. We observe a trend that the number of output tokens increases as the document size increases, similar to the trend observed in Figure 6 for the news summary dataset.

B Model Details

Table 4 shows the model details used in our experiments.

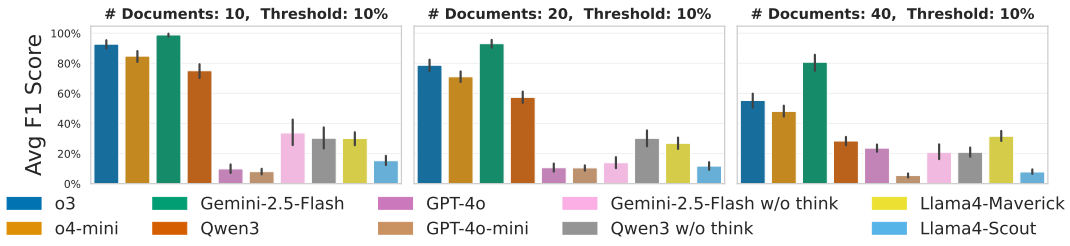
C Prompts

Figure 11 shows a prompt used for the DFM task in this paper.

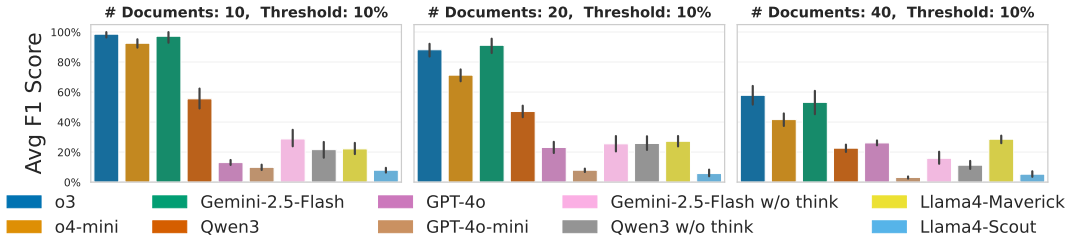
[1] https://openai.com/policies/services-agreement/ [Accessed: July 26, 2025]

[2] https://ai.google.dev/gemini-api/terms [Accessed: July 26, 2025]

[3] https://www.llama.com/llama4/license/ [Accessed: July 26, 2025]



(a) Resumes



(b) News summaries

Figure 8: F1 scores with various document sizes when $\theta = 0.1$.

Model	Size	Context	HuggingFace / API	License
o3 (OpenAI, 2025)	-	200k	`o3-2025-04-16`	OpenAI Service Terms[1]
o4-mini (OpenAI, 2025)	-	200k	`o4-mini-2025-04-16`	OpenAI Service Terms
Gemini-2.5-Flash (Comanici et al., 2025)	—	1M	`gemini-2.5-flash-preview-04-17`	Gemini API Additional Terms of Service[2]
GPT-4o (OpenAI, 2024)	—	128k	`gpt-4o-2024-08-06`	OpenAI Service Terms
GPT-4o-mini (OpenAI, 2024)	—	128k	`gpt-4o-mini-2024-07-18`	OpenAI Service Terms
Llama-4-Maverick (Meta, 2025)	400B	1M	`meta-llama/Llama-4-Maverick-17B-128E-Instruct`	Llama 4 Community License Agreement[3]
Llama-4-Scout (Meta, 2025)	109B	10M	`meta-llama/Llama-4-Scout-17B-16E-Instruct`	Llama 4 Community License Agreement
Qwen-3 (Team, 2025)	235B	128k	`Qwen/Qwen3-235B-A22B`	Apache license 2.0

Table 4: Models used in experiments. Model sizes are not publicly disclosed (-).

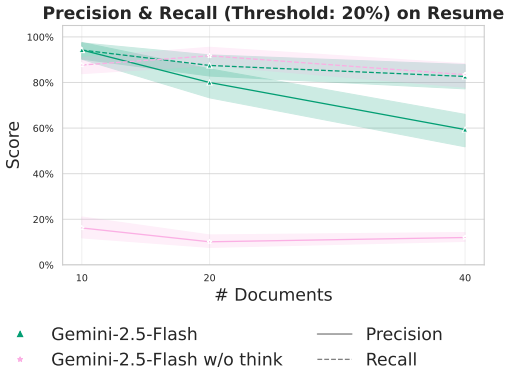


Figure 9: The precision and recall with various document sizes (Resume dataset).

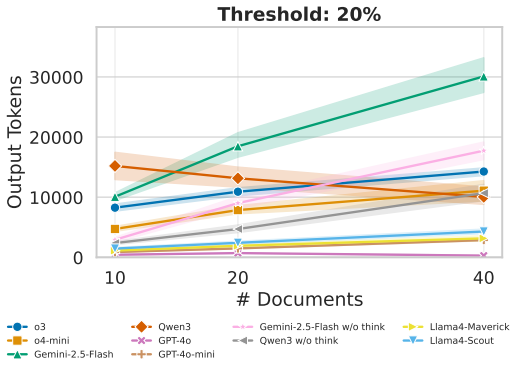


Figure 10: The average number of output tokens with various document sizes (Resume dataset).

 # Role
You are an expert AI assistant specializing in comparative resume analysis.

 # Objective
You will be given {num_documents} resumes. For each candidate, identify their "distinctive features" (such as skills, tools, or certifications) that are held by {distinctive_threshold} or fewer of the total candidates.

 # Instructions
- Identify features from each resume (e.g., programming languages, software tools, professional certifications, unique projects).
- Provide your thinking and reasoning process before listing the features.
- Count the occurrences of each feature across all resumes to determine which ones meet the "distinctive" criteria (appearing {distinctive_threshold} or fewer times).
- For each candidate, create a list of the distinctive features they possess.
- When listing features, use the exact wording as it appears in the resume. Do not summarize or rephrase.
- If a candidate has no qualifying distinctive features, return an empty list '[]'.
- Your output must be in valid JSON format.

 # Input Resumes:

{resumes}

Output Format ({num_documents} candidates)

```
{{
  "outputs": [
    {{
      "candidate_id": 1,
      "reasoning": "Your reasoning and analysis for candidate 1",
      "output": [
        distinctive_feature_1,
        distinctive_feature_2,
        ...
        ]
    }},
    ...
    {{
      "candidate_id": {num_documents},
      "reasoning": "Your reasoning and analysis for candidate {num_documents}",
      "output": [
        distinctive_feature_1,
        distinctive_feature_2,
        ...
        ]
    }}
  ]
}}
```

Figure 11: DFM task prompt template for the resume dataset. Variables {num_documents} and {distinctive_threshold} are replaced with the number of documents and the distinctive threshold, respectively. The {documents} variable is replaced with the actual documents.