On the Necessity of World Knowledge for Mitigating Missing Labels in Extreme Classification

Jatin Prakash[*†]
jatinprakash1511@gmail.com
New York University
New York City, NY, United States

Anirudh Buvanesh[*†]
anirudhb1102@gmail.com
Mila - Quebec Artificial Intelligence Institute
Montreal, QC, Canada

Bishal Santra
bishalsantra@microsoft.com
Microsoft Research
Bengaluru, Karnataka, India

Deepak Saini
desaini@microsoft.com
Microsoft Corporation
Mountain View, CA, United States

Sachin Yadav[†]
sachinyadav7024@gmail.com
Google DeepMind
Bengaluru, Karnataka, India

Jian Jiao
jian.jiao@microsoft.com
Microsoft Corporation
Bellevue, WA, United States

Yashoteja Prabhu
yprabhu@microsoft.com
Microsoft Research
Bengaluru, Karnataka, India

Amit Sharma
amshar@microsoft.com
Microsoft Research
Bengaluru, Karnataka, India

Manik Varma
manik@microsoft.com
Microsoft Research
Bengaluru, Karnataka, India

Abstract

Extreme Classification (XC) aims to map a query to the most relevant documents from a very large document set. XC algorithms used in real-world applications typically learn this mapping from datasets curated from implicit feedback, such as user clicks. However, these datasets often suffer from missing labels. In this work, we observe that *systematic* missing labels lead to missing knowledge, which is critical for modelling relevance between queries and documents. We formally show that this absence of knowledge is hard to recover using existing methods such as propensity weighting and data imputation strategies that solely rely on the training dataset. While Large Language Models (LLMs) provide an attractive solution to augment the missing knowledge, leveraging them in applications with low latency requirements and large document sets is challenging. To mitigate missing knowledge at scale, we propose SKIM (**S**calable **K**nowledge **I**nfusion for **M**issing Labels), an algorithm that leverages a combination of Small Language Models (SLMs), e.g., Llama-2-7b, and abundant unstructured meta-data to effectively address the missing label problem. We show the efficacy of our method on large-scale public datasets through a combination of unbiased evaluation strategies, such as exhaustive human annotations and simulation-based evaluation benchmarks. SKIM outperforms existing methods on *Recall@100* by more than 10 absolute points. Additionally, SKIM scales to proprietary query-ad retrieval datasets containing 10 million documents, outperforming baseline methods by 12% in offline evaluations and increasing ad click-yield by 1.23% in an online A/B test conducted on Bing Search. We release the code and trained models at: github.com/bicycleman15/skim

CCS Concepts

• **Computing methodologies → Supervised learning by classification**.

Keywords

extreme classification, missing labels, large-scale retrieval

ACM Reference Format:
Jatin Prakash, Anirudh Buvanesh, Bishal Santra, Deepak Saini, Sachin Yadav, Jian Jiao, Yashoteja Prabhu, Amit Sharma, and Manik Varma. 2025. On the Necessity of World Knowledge for Mitigating Missing Labels in Extreme Classification. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 26 pages. https://doi.org/10.1145/3690624.3709290

1 Introduction

Extreme Classification (XC) addresses the challenge of mapping a query to the most relevant subset of documents from a very large set of documents. XC algorithms have demonstrated impressive performance across a wide range of applications, including document tagging [9, 104], product recommendation [22, 23, 65], and search and advertisement [40, 69]. However, these datasets are prone to *missing labels*, as it is impossible to annotate all query-document pairs when the document space is vast, consisting hundreds of millions or even billions of documents. The XC models trained on such datasets with missing labels may fail to accurately capture query-document relevance.

The extreme classification (XC) community has developed two primary approaches to address the challenge of missing labels. The first approach, propensity-based methods [39, 77, 97, 99], constructs unbiased loss functions by reweighting the original loss with query-document observational probabilities (i.e. propensities). The second

[*]Equal contribution
[†]Work done while at Microsoft Research; Correspondence to: jatin.prakash@nyu.edu, anirudh.buvanesh@mila.quebec, yprabhu@microsoft.com

KDD '25, August 3–7, 2025, Toronto, ON, Canada

ACM ISBN 979-8-4007-1245-6/25/08
https://doi.org/10.1145/3690624.3709290

approach encompasses naive imputation methods [17, 50], which aim to infer missing labels using various techniques. However, both these approaches are limited by their heavy reliance on large-scale but potentially biased training data, as explained later in this paper.

In this work, we revisit the problem of missing labels in XC from the novel perspective of *world knowledge*. Most XC tasks are knowledge-intensive and require a deep understanding of named entities such as people, brands, or locations as well as other forms of knowledge. For example, in Figure 1, determining the relevance of a document *"medical.net/what-are-genes"* to the query *"what is an exon?"* requires the understanding that *exons* are an integral component of *genes*. Such world knowledge becomes increasingly important in short-text based XC tasks where queries and documents are pithy.

XC models are usually trained using supervised learning from relevant query-document training pairs. As a result, the knowledge they acquire is restricted to this training dataset too. However, the vast and diverse nature of world knowledge presents a significant challenge: limited human-annotated ground truth datasets fall short of comprehensively covering the long tail of world knowledge required for optimal performance. We refer to the special case of missing labels in XC, where all the labels containing some specific knowledge go missing together, as the phenomenon of "systematic missing label bias." We rigorously demonstrate, both theoretically and empirically, that traditional supervised learning techniques that rely solely on the training dataset tend to lose world knowledge and, furthermore, existing solutions to missing labels are inherently limited in their ability to recover such missing knowledge. To address these limitations, it is necessary to go beyond the world knowledge encoded in the training datasets.

Recently, Large Language Models (LLMs) have been gaining popularity. These LLMs have been demonstrated to be effective as knowledge bases [37] which store vast world knowledge in their large number of model parameters. Due to their superior language understanding, they can also effectively model users' relevance notions[92]. Naturally, LLMs offer a potential solution to bridge this missing knowledge gap in retrieval datasets. However, it is challenging to utilize them in applications having large output spaces as they fail to scale and satisfy the latency requirements of XC applications. A promising alternative is to utilize the more scalable Small Language Models (SLMs), but these lack sufficient parametric world knowledge making them unreliable for accurate generation [46].

To address these challenges, we propose SKIM (**S**calable **K**nowledge **I**nfusion for **M**issing Labels), an algorithm that combines the utility of raw document metadata and SLMs to mitigate the problem of missing knowledge in extreme classification at scale. We observe that in XC datasets, documents often include readily available metadata containing useful knowledge. However, this metadata is typically noisy, making it difficult to utilize effectively [67]. Thus, our approach consists of two key stages: (i) *diverse query generation*: using unstructured metadata text, we generate diverse synthetic queries that cater to different knowledge items within the document, (ii) *retrieval-based mapping*: we employ a retrieval model to map the generated synthetic queries to the space of the train queries in the dataset. Through this process, SKIM extracts only the relevant knowledge pertinent to the retrieval

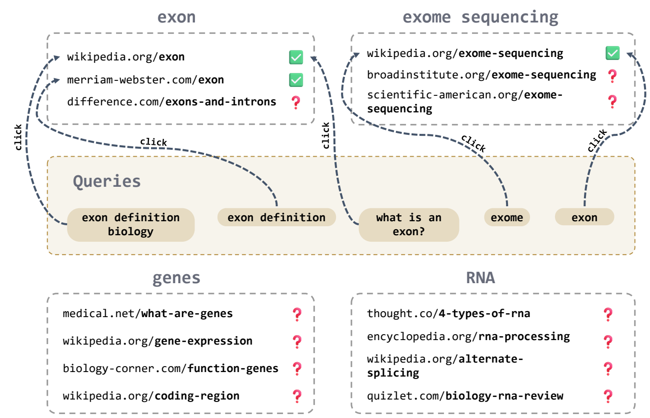


Figure 1: Connections between queries and documents define the knowledge available in a retrieval dataset. In the above example, the document concept *exome sequencing* is connected to the query "exon" through a user click, encoding the knowledge that concepts *exon* and *exome sequencing* are related. The document concept *exon* is not directly connected to query "exome" but this relationship can possibly be learnt (see Table 13 in appendix). However, the relevance of *exon* to *genes* or *RNA* is impossible to learn through this dataset because there are no connecting clicks providing this knowledge, that is, those connections (documents) are *systematically* missing for *exon*. (Note: ✅ implies *relevant & clicked*, ❓ implies *relevant & missing*)

task from unstructured metadata and augments it to the existing training dataset, making it generally applicable to any retrieval model. SKIM efficiently scales to datasets containing millions of documents, while maintaining the latency necessary for XC tasks.

In this work, we additionally observe that datasets collected from click logs, like their training sets, have test sets corrupted by the same missing-label bias, rendering existing evaluation benchmarks unreliable. To address this, we evaluate SKIM and other baselines using a combination of more trustworthy alternatives: (i) small-scale test sets with exhaustive human annotations, (ii) simulated setups designed to replicate biases encountered in real-world industrial settings, and (iii) live A/B tests. We test the efficacy of SKIM not just on large-scale public datasets, but also on a real-world query-ad keyword retrieval dataset, where SKIM scales to 10 million ad keywords (documents). SKIM outperforms contemporary methods by 12% in offline evaluations and improves ad click-yield by 1.23% in an online A/B test conducted on the Bing search engine.

In summary, this work makes the following contributions:

- Studies a novel and important connection between the world knowledge and the missing labels in XC, and introduces the notion of systematic missing label bias.
- Establishes, through rigorous theoretical arguments, the inefficacy of standard debiasing techniques such as propensity-scoring and naive label imputation in mitigating systematic missing labels.

- Proposes SKIM, a highly accurate and scalable method, to effectively mitigate the systematic missing labels problem by leveraging the knowledge present in LLMs/SLMs and unstructured metadata text.
- Reliably estimates unbiased performance of SKIM and competing baselines using a diverse set of evaluation strategies, including human annotated test set, controlled simulation and live A/B test conducted on a popular search engine. We additionally perform extensive ablations to support our design choices.
- We open-source our code, trained XC models, finetuned SLMs, generated synthetic datasets using SKIM, LLM prompts & finetuning LLM responses at: github.com/bicycleman15/skim

2 Related work

Extreme classification (XC):

Recent advancements in XC primarily fall into two categories: one-versus-all classifier-based methods [39, 76] and dual encoder approaches [23, 33, 34, 40, 49]. Over time, XC methods have evolved significantly, transitioning from sparse feature-based models [6, 9–11, 15, 38, 39, 48, 74–76] to those leveraging deep networks [23–25, 66, 68, 104]. These advancements include end-to-end model training [40], incorporating label-text features using text-encoder models [58, 59, 78], and employing effective negative sampling strategies [23, 32, 44, 82, 83, 101]. In this work, we take a data-centric approach to enhance the performance of XC methods, ensuring compatibility with any XC architecture.

Missing label bias: XC training datasets suffer from missing label bias [39], stemming from selection [63], position [19], exposure [53, 106], and inductive biases [18] prevalent in retrieval applications. This bias adversely affects the accuracy of model training.

The most widely adopted approach to addressing the missing label problem in XC is propensity-based learning [39, 77, 91, 97, 99]. Introduced by Jain et al. [39], this approach involves reweighting the loss function using per-label propensity scores, which estimate the likelihood of a label being missing. However, Schultheis et al. [91] identified key limitations of this method, including: (i) its reliance on query-independent propensity models and (ii) the lack of auxiliary data for estimating propensity model parameters across datasets.

While propensity-based methods have led to some improvements, we show that they struggle to predict relevant query-label pairs outside the training distribution's support (see Table 13).

Teacher Models and Data Augmentation: Another line of research focuses on enhancing retrieval models by augmenting training datasets with additional resources, such as teacher models, query or document metadata [5, 16, 27, 43, 67, 69, 78, 86, 103]. For instance, LEVER [17], RocketQA [103] and Gandalf [50] addresses this by relying on pretrained retrievers or leveraging label correlations statistics. However, their continued reliance on limited and biased training data affects their ability to resolve the missing knowledge problem effectively.

Other methods [69, 87, 103] implicitly attempt to augment knowledge by retrieving candidate documents from external corpora, which are then scored using a teacher model. However, the retrieval process often depends on small, pretrained encoders with limited parametric knowledge (*e.g.,* 6 layer `msmarco` pretrained DistilBERT), restricting their ability to discover novel candidates. Furthermore, teacher models themselves may lack the knowledge necessary to score candidate documents accurately [17, 43, 86]. Our approach overcomes these limitations by leveraging unstructured metadata associated with documents and employing small language models (SLMs) to generate diverse candidate documents directly. Unlike retrieval-based methods, our approach taps into the implicit world knowledge encoded in the metadata, enabling the discovery of novel and relevant candidates that improve retrieval performance.

Recognizing the potential benefit of leveraging metadata, some recent methods in XC [67, 69] have also started utilizing external document or query metadata. However, these approaches require metadata in specific, structured forms (*e.g.,* curated links between query and external corpora), which are rarely available, and susceptible to the missing label bias itself. In contrast, our method can work seamlessly with unstructured metadata, making it more versatile and broadly applicable compared to prior methods that depend on structured metadata.

Language Models and Synthetic Data: LLMs have recently gained traction for generating data to train task-specific models [52], smaller language models (SLMs) [31, 36, 61, 84, 85], and multimodal models [56]. These approaches address challenges such as data scarcity in low-resource settings, offering an effective solution to improve model performance in such scenarios.

However, directly employing LLMs for dataset generation can be prohibitively expensive, especially for XC datasets, which require large-scale data. Methods that utilize SLMs [43, 86] for retrieval dataset generation face their own challenges due to the limited parametric knowledge available in smaller models.

In parallel, efforts have explored leveraging SLMs directly for retrieval tasks. Approaches such as fine-tuning these models for retrieval [60, 70, 72], or rephrasing queries and documents before retrieval [94], have shown promise. Nevertheless, these methods struggle to meet the low latency requirements critical for XC tasks.

Our method addresses these limitations by augmenting the limited parametric knowledge of SLMs with relevant metadata during the generation of novel queries for a document. Additionally, we ensure scalability for XC workloads by *distilling* the world knowledge from the metadata into an efficient retriever. This approach maintains low inference costs while catering to the missing knowledge problem.

3 Theory: Necessity of External Knowledge

This section studies the phenomenon of *systematic* missing label bias in XC. It characterizes the nature, origin and properties of the systematic missing labels, and theoretically demonstrates that the classical debiasing techniques of XC such as propensity scoring and naive imputation are unable to effectively address such biases.

3.1 Preliminaries

Extreme Classification: Extreme Classification or XC involves learning a statistical model $\mathcal{M}$ that takes a query $\mathbf{x} \in \mathcal{X}$ as input and predicts relevant subset of documents $\mathbf{y} \in \{0, 1\}^L$ from a set of L (a large number) documents with the lth document deemed relevant to

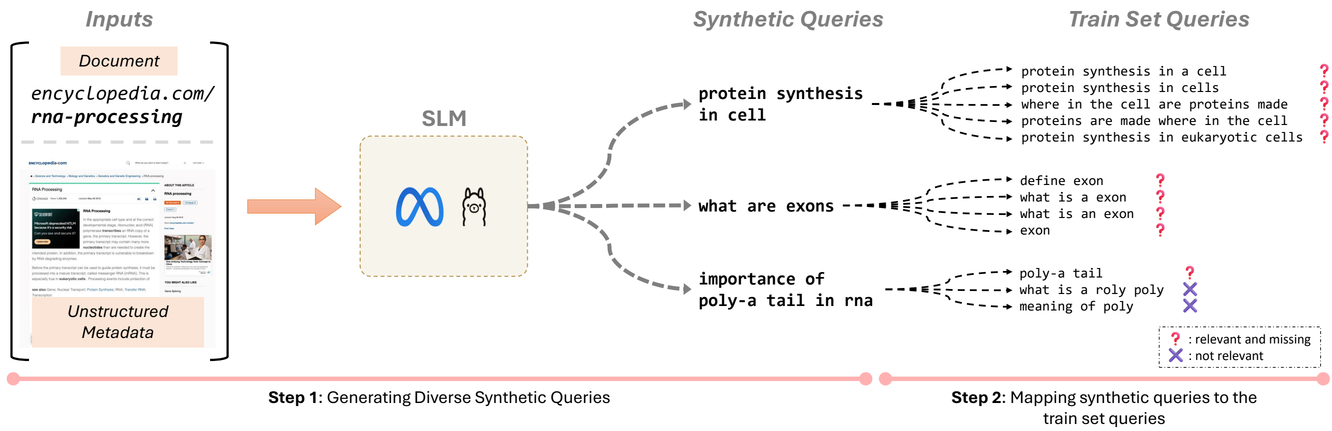


Figure 2: Steps of the SKIM algorithm depicting how we bridge the missing knowledge in biased training datasets. In Step 1, for a URL (document) "*encyclopedia.com/…/rna-processing*", the finetuned SLM generates diverse synthetic queries spanning concepts like *protein synthesis, exons, poly-a tail* etc using the available unstructured meta-data (see Figure 4 in appendix). In Step 2, a retriever is used to increase the coverage of the chosen document to relevant train queries through these synthetic queries, e.g. synthetic query "*what are exons*" is mapped to similar train queries like "*define exon*" and "*exon*" which were missing for the document. The retriever additionally filters out irrelevant train queries for the document using the similarity threshold τ.

$\mathbf{x}$ if $y_l = 1$ and irrelevant otherwise. The model $\mathcal{M}$ is trained by supervised learning on a large training set $\mathcal{D} = \{\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N, \{\mathbf{z}_l\}_{l=1}^L\}$ where $\mathbf{x}_i, \mathbf{z}_l$ are the textual features of ith query and lth document respectively and $\mathbf{y}_i$ are the ground truth documents for $\mathbf{x}_i$. Often, $\{\mathbf{y}_i\}_{i=1}^N$ contains many false negatives where a document l is indeed relevant to query i but is labelled as irrelevant (*i.e.* $y_{il} = 0$). This is the well-known problem of missing label bias in XC [39, 91]. Training $\mathcal{M}$ on data with missing labels can lead to inferior model quality and yield inaccurate predictions when deployed in real-world applications. This paper revisits the missing label bias from the perspective of missing world knowledge.

Training data collection: As the number of queries and documents in a typical XC application can range in millions or more, exhaustive human annotation of all query-document pairs becomes infeasible. Consequently, most applications rely on implicit feedback from users to cost-effectively generate training data at scale. For example in a search application, when a user asks a query $\mathbf{x}_i$ sampled from a distribution $P_x = \mathbb{P}(\mathbf{x})$, a serving algorithm processes and predicts some documents which are presented to the user and clicked on when found to be relevant to the query. The data is aggregated over a finite time period, and the training dataset $\mathcal{D}$ is constructed from the set of all user-entered queries, available documents and the user clicks from this period. Click-based training data often suffers from several inadequacies. First, user clicks are sparse as a single document tends to be clicked for a typical user query. Let the probability that a document $\mathbf{z}$ gets clicked for a query $\mathbf{x}$ be denoted as $P_{z|x} = \mathbb{P}_{\text{click}}(\mathbf{z}|\mathbf{x})$. Second, user clicks are inaccurate and subject to exposure and position biases. In exposure bias, due to the inaccuracies of the serving algorithm, some relevant documents are not displayed for a query, and therefore not clicked. In position bias, the users click on higher-ranked documents despite being potentially less relevant compared to some other documents. These biases manifest themselves in the training data as missing labels. For simplicity, we assume a uniform value of $1 - B$ for exposure bias. Given this, the true probability that the document $\mathbf{z}$ is relevant to query $\mathbf{x}$ becomes $\mathbb{P}(y_{xz} = 1|\mathbf{x}, \mathbf{z}) = P_{z|x}/B$.

3.2 Systematic Missing Label Bias

World knowledge: In many XC tasks, the query and document texts, on their own, do not contain sufficient information to make relevance decisions. Therefore, the users draw upon additional world knowledge for their clicks. Typically, the users derive such knowledge either from their personal world knowledge or from reading additional meta-data provided with the documents. An important property of world knowledge is that it is not easily compressible across training samples. For example, in Figure 2, a knowledge statement that "*rna* is related to *protein-synthesis*" does not straight-forwardly entail the statement "*exon* is related to *rna* or *genes*". Due to this, solving a large-scale XC task typically requires access to vast amounts of knowledge associated with a long-tail distribution. For precise understanding, we formalize such intuitions about world knowledge into concrete mathematical statements below.

We assume that the world knowledge required for an XC task can be posed as a large and discrete set of knowledge items $\mathcal{K} = \{k_m\}_{m=1}^K$. These items could correspond to statements such as those in the previous para; but the following analysis remains agnostic to the actual definition of set $\mathcal{K}$, and only assumes that $\mathcal{K}$ satisfies the following conditions:

Postulate 1. ***Externality of*** $\mathcal{K}$: *The world knowledge is not derivable from and is statistically independent of query and document*

texts:
$\{k_m\}_{m=1}^K \perp\!\!\!\perp \{x_i\}_{i=1}^N, \{z_l\}_{l=1}^L$

Postulate 2. ***Incompressibility of $\mathcal{K}$:*** *The world knowledge items are not derivable from and are statically independent of each other:*
$k_m \perp\!\!\!\perp \mathcal{K} \setminus k_m \quad \forall m \in [1, \cdots, K]$

As in the examples discussed before, typically, the users require only a small amount of additional knowledge to accurately judge any given query-document pair. We assume a simplified setting based on this as follows:

Postulate 3. ***Sparsity of $\mathcal{K}$:*** *The relevance of any query-document pair can be fully determined by using exactly one item from the knowledge set:*
$y_{il} = R(\mathbf{x}_i, \mathbf{z}_l, k_{\mathbf{x}_i,\mathbf{z}_l}) \in \{0, 1\}$ *where R is the deterministic true relevance function and $k_{\mathbf{x}_i,\mathbf{z}_l}$ is the knowledge item utilized by the pair. For notational convenience, let $D_m = \{(\mathbf{x}, \mathbf{z}) \in \mathcal{X} \times \mathcal{Z} : k_{\mathbf{x},\mathbf{z}} = k_m\}$ denote the space of query-doc pairs which depend on the item m.*

Note that, the three postulates mentioned above hold reasonably true in the setting where queries and documents are both short-text, which is the main focus of this paper. However, these may not hold in general retrieval scenarios (e.g. long-text queries or documents), and more realistic settings can be explored in the future.

Systematic missing labels: The necessity of world knowledge for judging relevance applies not only to human users but also to the XC algorithms used to serve the predicted documents. The XC models are typically expected to acquire such knowledge during their supervised training from a dataset $\mathcal{D}$. For this, the training data needs to have adequate coverage of $\mathcal{K}$ among its clicked (*i.e.* positive/relevant) query-document pairs. Conversely, if some knowledge item $k \in \mathcal{K}$ is completely missing in the training data, then under the above-mentioned theoretic assumptions, k cannot be learned by the supervised model even when trained with a large dataset. This is formalised by the following lemma:

Lemma 1. *If $y_{il} = 0 \quad \forall(\mathbf{x}_i, \mathbf{z}_l) \in D_m$, then for any test pair $(\mathbf{x}, \mathbf{z}) \sim D_m$, $R(\mathbf{x}, \mathbf{z}, k_m) \perp\!\!\!\perp \mathcal{D}$ where $\mathcal{D} = \{\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N, \{\mathbf{z}_l\}_{l=1}^L\}$ is the training dataset.*

We refer to such a missing label phenomenon, where all the relevant query-document pairs corresponding to some knowledge item go missing in the training ground truth, as the problem of **systematic missing labels**. Systematic missing labels lead to loss of crucial knowledge which cannot be re-constructed by any supervised model (including extreme classifiers) trained on $\mathcal{D}$. As a result, such models will be unable to reliably predict any relevant query-document pair that depends on the missing knowledge k_m, as proven next:

Corollary 2. *If $y_{il} = 0 \quad \forall(\mathbf{x}_i, \mathbf{z}_l) \in D_m$, then for any test pair $(\mathbf{x}, \mathbf{z}) \sim D_m$, $R(\mathbf{x}, \mathbf{z}, k_m) \perp\!\!\!\perp \mathcal{M}$; where $\mathcal{M}$ is a predictive model trained deterministically from $\{\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N, \{\mathbf{z}_l\}_{l=1}^L\}$.*

When the missing k_m belongs to the long-tail of $\mathcal{K}$, the above result also holds true for propensity-scored approaches [39, 77], as they typically train on just $\mathcal{D}$ and, a small unbiased test set $\mathcal{D}'$ which is unlikely to contain pairs from k_m. Similarly, this also applies to naive imputation techniques [17, 50] that ultimately leverage the ground truth or another supervised model learnt from the ground truth. Note that these techniques can still be useful for addressing the non-systematic missing labels that are not associated with tail knowledge, but that is not the focus of this paper.

3.3 Lower Bound on Model Performance

While the previous part presents systematic missing labels as a theoretical construct, they are indeed real and frequent in many practical XC applications requiring vast and long-tail knowledge (for example see Figure 3 in Appendix). There are two key reasons behind this. First, long-tail knowledge items are hard to comprehensively cover, even with a large training set, due to the inherent stochasticities of the query sampling process. Second, user clicks are also affected by document exposure bias which further reduces the amount of clicks on the long-tail.

During click-based training set generation, the marginal probability of sampling a relevant pair $(\mathbf{x}, \mathbf{z}) \in D_m$ associated with a knowledge item m is $p_m = \sum_{(\mathbf{x},\mathbf{z}) \in D_m} P_x P_{z|x}$. Note that $\sum_{m=1}^K p_m = 1$. *W.l.o.g* assume that the indices $\{1, \cdots, K\}$ of knowledge items are sorted according to decreasing marginals p_m. Also, denote the survival function $\bar{F}_m = \sum_{m'=1}^m p_{m'}$. Now, a long-tailed distribution is typically expected to have a significant portion of aggregate density in the tail portion, *i.e.* for some large m, $\bar{F}_m$ is reasonably large even though p_m is very small. The below theorem establishes a lower bound on the amount of systematic missing labels introduced in the training data as a function of the amount of knowledge tail and exposure bias.

Theorem 3. *Let n be the number of clicked query-doc pairs sampled for the training data, B be the exposure bias while sampling, and $\bar{F}_m, p_m$ be defined as above. Then, for any $0 \le \delta \le 1$, with probability at least δ, at least $\max_{m=1}^K \frac{\bar{F}_m}{B} e^{-B.p_m.n} - \frac{1}{2\sqrt{2n}} (\log \frac{2K}{1-\delta}^{\frac{3}{2}})$ fraction of the true relevant label distribution will correspond to the systematic missing label region and therefore will be irrecoverable by supervised training.*

Note, from the above theorem, that (1) the error can be significant when p_m is of the order $\frac{1}{n}$ which can happen when the knowledge grows with the training set size, (2) the error increases with the exposure bias $1 - B$. Proof is in the Appendix.

Systematic missing labels are a serious problem because, not only are they hard to mitigate with biased training data alone, but are also subject to vicious feedback loops in the retrieval systems where errors owing to missing knowledge get absorbed by future serving models and get amplified with every model re-training event if not addressed properly.

4 Method

In the previous section, we established that the implicit-feedback based training data is insufficient for bridging the existing knowledge gaps. To tackle this issue, this section develops a novel algorithm that leverages the external world knowledge beyond what is available in the training dataset.

4.1 SKIM: Scalable Knowledge Infusion for Missing Labels

Overview. The recent advancements in LLMs have demonstrated that they can act as knowledge bases [37] and can accurately predict user search preferences [92] upon adequate prompting. These LLMs offer a natural solution to mitigate knowledge gaps. A naive strategy is to label every query-document pair in the dataset with an LLM annotator. However, this can become prohibitively expensive, requiring $O(t_{LLM} \cdot N \cdot L)$ calls to a slow and bulky LLM like GPT4, with both N (number of queries) and L (number of documents) ranging in millions. Instead, we propose an alternative strategy of directly generating the relevant queries for a document thus reducing the cost to $O(t_{LLM} \cdot L)$. Additionally, we develop SKIM technique to replace GPT4 with a smaller and faster Language Model (SLM) while retaining the quality of query generation.

SKIM: SKIM stands for **S**calable **K**nowledge **I**nfusion for **M**issing Labels. The SKIM technique leverages an SLM like LLAMA-7B or Phi-1.5B to impute systematic missing labels. The imputation process itself involves two successive steps: (i) SLM based generation of diverse synthetic queries that are representative of the missing knowledge for a document, (ii) Projection of these synthetic queries onto actual user queries in the training set using an embedding-based nearest neighbour search procedure. In this way, SKIM completes the ground truth matrix over the original set of queries and documents with these additional relevant pairs that correspond to missing knowledge. The use of SLM, however, brings up two technical limitations: (i) SLMs have limited world knowledge owing to their smaller model capacity, (ii) SLMs are more prone to hallucinations (Fig. 6 in Appendix). SKIM addresses these limitations through strategies of raw metadata augmentation and LLM-guided distillation respectively. We first present the two-step augmentation procedure from the perspective of an LLM. Thereafter, we discuss the SLM-specific optimizations that make augmentation more scalable while retaining the quality of generated data.

Step 1. Generating diverse synthetic queries. We prompt a LLM to generate diverse synthetic queries given a document. Given that knowledge items are often incompressible, diversity in query generation is important to ensure coverage of various knowledge items. Generating many queries for a document is specifically suited for the XC setting where the model trains a separate classifier for each document thus requiring sufficient relevant query-document pairs for each classifier.

Due to the extensive parametric world knowledge of LLMs, they are capable of generating queries that can address existing knowledge gaps in typical XC training datasets. For instance, in the LF-Orcas-800K dataset, which involves matching user queries to relevant URLs, a LLM like GPT4 generates synthetic queries for a URL (document) such as *encyclopedia.org/.../rna-processing* that span diverse knowledge items relating to *genes*, *exons* and, *DNA*. To further improve the knowledge coverage, one could also supply any available metadata along with the document to LLM. We instruct the LLM using a well-crafted prompt that describes the task and provides high-quality in-context examples that promote diversity and accuracy in synthetic query generation. For our implementation, we use GPT-4 [3] as the LLM. The prompts used are provided in the Appendix E.1. This step is repeated for every document, resulting in an asymptotic time complexity of $O(t_{LLM} \cdot L)$. In practice, however, this process would be costly and cannot scale to XC datasets containing millions of documents as noted earlier. Therefore, we discuss a scalable version of Step 1 in the subsequent subsection 4.2, that scales to XC datasets and is accurate at the same time.

Step 2. Mapping synthetic queries to train set queries. While the synthetic queries are relevant and diverse, they may not match the distribution of real users' queries. Assuming that the number of queries is vast, to enforce the true distribution over queries, we map them to real queries. Specifically, we use the generated synthetic queries from the previous step and map them onto the train queries, that is, for every synthetic query corresponding to a document, we take the nearest neighboring train queries in the train set and consider them as relevant train queries for this document. This step also helps to increase the number of relevant queries for each document. For instance, a synthetic query like "*what are exons*" gets mapped to multiple training queries such as "*define exon*", "*what is an exon*", and "*exon*" which inform the XC model of the different query variations corresponding to the same knowledge item thereby making it more robust to test-time queries. To ensure that only high-quality (train query, document) pairs are retained, we only consider neighboring queries for a synthetic query that have a cosine similarity greater than τ, which is tuned as a hyper-parameter. Alternatively, one may potentially use a sophisticated filter model for this step, but we leave it as future work. For nearest neighbor search, we employ an Approximate Nearest Neighbor Search (ANNS) algorithm [62], which can be run efficiently on CPUs and can scale to large datasets. We use a finetuned dual encoder [23] on the given training dataset as the embedding space in which we calculate nearest neighbors. The resulting synthetic (train query, document) pairs are finally combined with the original training pairs to create the final, knowledge-augmented training set.

The above two steps complete the basic components of our method. As previously noted, Step 1 is reliant on LLMs and is therefore computationally expensive. Thus, we now discuss how to scale-up Step 1 for practical XC workloads.

4.2 Scaling synthetic query generation in SKIM

To scale synthetic query generation to large XC datasets, we employ SLMs for query generation. Naively prompting a pretrained SLM is observed to yield inferior generation quality (see ablation in Appendix G, Figure 6). This is owing to (i) a lack of parametric world knowledge compared to LLMs, and (ii) a pre-trained SLM is prone to hallucinations (see Fig. 6 in appendix). To address the first issue, we supplement the SLM with raw metadata that provides additional world knowledge in a noisy form. Such metadata is readily available for many XC tasks. For example, in LF-Orcas-800K where documents are URLs, we can use webpage text as associated metadata; in query-ad keyword retrieval datasets, we can use the ad landing page for the keyword; and for Wikipedia titles, we can use their article content etc. To address the second issue, we perform teacher-to-student distillation from an LLM onto an SLM.

Task-specific distillation. To collect distillation data for SLM, we instruct the large language model (LLM) using the same prompt developed in Step 1. Since the task will ultimately be performed by an SLM, we explicitly instruct the LLM to *only* utilize the knowledge present in the metadata for synthetic query generation, and to *not* use its own parametric world knowledge. This ensures that the SLM can solve this task using the available metadata[1] only. Additionally, this inherently demands the LLM (and eventually the distilled SLM) to filter out irrelevant text from the raw metadata text (refer to Figure 4 in the appendix).

For curation of distillation data, we use GPT-4, collecting approximately 50K responses per dataset, generating ∼10 synthetic queries per example/document. We then distill this particular task to an SLM via vanilla supervised fine-tuning (SFT) using Low-Rank Adaptation (LoRA). Post fine-tuning, the SLM is observed to yield generation quality that is comparable to an LLM generation quality (see Fig. 6 in appendix). Our experiments use two SLMs: Llama2-7B [93] for the primary experiments, and Phi1.5-1.3B [42] for ablation studies. Detailed information on SFT and LoRA hyper-parameters, distillation time, and hardware used is provided in Appendix E.

Large-scale inference. Once we have the distilled SLM, we perform large-scale inference on all the documents (alongside their associated metadata) to generate synthetic queries. With these generated synthetic queries for each document, we can then proceed to the original Step 2 discussed in the previous section. Refer to Appendix E for details on SLM inference time and hardware. In this way, we are able to perform Step 1 in SKIM accurately at scale. This reduces the generation time from several weeks/months to under a day for XC datasets, and at the same time, maintaining high-quality of generation similar to that of LLMs (see Fig. 6 in appendix).

5 Experimental Results

We now discuss the experimental setup, including datasets, evaluation and key observations. We also assess the real-world applicability of SKIM through offline and online tests on a search engine.

5.1 Setup

Datasets: We benchmark SKIM on two public XC datasets, LF-Orcas-800K [20, 26] and LF-WikiTitlesHierarchy-2M [17].

LF-Orcas-800K: This dataset involves mapping short user search queries to URLs of web pages that answer those queries. The training set is curated from click logs of the Bing Search engine [20], thus making it prone to missing labels. To ensure unbiased evaluation, we build a test set using human-labelled queries from the TREC-19 and TREC-20 Deep Learning competitions [21]. Web-page text for a URL (document) is used as unstructured metadata.

LF-WikiTitlesHierarchy-2M: This dataset involves mapping Wikipedia titles to their categories. Since Wikipedia categories are human-annotated, the degree of missing labels is relatively less severe. Therefore, we assume this dataset to be relatively unbiased and simulate a controlled click bias (only in training set) using a pre-trained `msmarco` model [2]. Specifically, only those documents that appear in the top-K ($K = 200$) predictions of the pre-trained `msmarco` model are considered in the biased training set. Note that our design choices are informed by production settings where only some documents predicted by some deployed model are shown to the user. The article text corresponding to the Wikipedia Title (query) is used as metadata. More details related to datasets and simulation are in Appendix C.

Baselines: We compare SKIM against two classes of baselines: those that do not use external knowledge and those that do. Within baselines without access to external knowledge, we compare to the propensity-based approach: Inverse Propensity Scoring (IPS) [39, 77], and imputation methods like Gandalf [50] and LEVER [17]. To explore a combination of propensity and imputation methods, similar to doubly robust techniques [55], we compare against a baseline that combines IPS with LEVER. Finally, for methods having access to external knowledge, we evaluate SKIM against a slightly modified version of UDAPDR [86], an SLM-based rewriting method tailored for our use case. The above methods are applied on two state-of-the-art base XC models: one-vs-all extreme classifier Renée [40] and a dual encoder DEXML [34]. More implementation details are in App. D.

Evaluation: Since unbiased evaluation is a hard problem as pointed in Schultheis et al. [91], we employ a mixture of evaluation methods to reliably judge model performance. Our main test sets correspond to an unbiased setting: evaluation on human annotated test sets in case of LF-Orcas-800K and evaluation on the original test set in LF-WikiTitlesHierarchy-2M. Since we focus on improving retrieval performance, we evaluate our method and baselines using $Recall@K(R@k)$ with $K = 25, 100$ i.e. higher values of K.

5.2 Main Results

Now we discuss main observations from our results in Table 1.

SKIM outperforms state-of-the-art XC methods. On both public datasets, SKIM, when used with any XC model, outperforms all baselines on unbiased test sets. In LF-Orcas-800K, where the training dataset exhibits real-world missing label bias, SKIM outperforms the closest baseline by an average of **4.68** absolute points in R@100 and **3.60** absolute points in R@25. Similarly, in LF-WikiTitlesHierarchy-2M, SKIM outperforms the closest competitor by **8.27** absolute points in R@100.

Importance of World Knowledge. All baseline methods (except UDAPDR) do not incorporate world knowledge, and thus suffer from inferior performance, with their results being in similar ranges (they have R@100 around 40% in LF-Orcas-800K). Methods such as LEVER [17] and Gandalf [50] rely on biased training data itself for imputation, thereby lacking world knowledge, crucial for predicting missing knowledge items. While UDAPDR [86] attempts to incorporate external knowledge by using an SLM, the observed improvements are sub par due to its limited parametric knowledge.

Incorrect Conclusions from Standard Offline Test Sets. Table 11 (found in appendix) reports the results on the biased test set on the dataset LF-WikiTitlesHierarchy-2M. We observe that on DEXML, methods that do not incorporate external knowledge have similar R@100 (∼43) on the biased test set, while SKIM has a R@100 of 37.64. However, on the unbiased test set, SKIM outperforms the nearest baseline by > 8 absolute points. Similar observations hold true in case of baselines using Renée, where Renée + LEVER achieves the highest biased R@100, but has unbiased R@100 significantly inferior to Renée + SKIM (more than 13 points). This can

[1]Note that we do not process the metadata text in any form. It is unstructured and can potentially contain irrelevant text about the document (see Figure 4 in Appendix)
[2]https://huggingface.co/sentence-transformers/msmarco-distilbert-base-v4

Table 1: Recall on unbiased test sets using two base XC models: Renée and DEXML. When compared to methods that do not use any external information (Gandalf, LEVER, LEVER + IPS), SKIM outperforms the closest baseline by 10.16 points in $R@100$. For additional metrics please refer Tables 9, 11 in the supplementary material.

	LF-Orcas-800K			LF-WikiHT-2M		
Method	**R@10**	**R@25**	**R@100**	**R@10**	**R@25**	**R@100**
DEXML	21.77	28.32	39.08	8.27	10.87	15.25
+ Gandalf	22.39	29.87	40.52	8.23	10.73	15.01
+ LEVER	21.70	29.26	40.45	7.57	9.89	13.78
+ IPS	22.31	29.14	39.42	8.10	10.92	15.92
+ LEVER + IPS	23.47	30.92	41.04	6.82	9.03	12.92
+ UDAPDR	26.42	33.71	47.70	6.83	10.27	17.15
+ SKIM (Ours)	**26.99**	**36.90**	**49.60**	**11.13**	**16.59**	**25.42**
Renée	21.75	28.22	38.42	7.89	9.95	13.07
+ Gandalf	20.89	27.58	34.66	8.46	11.08	15.30
+ LEVER	22.58	28.40	39.52	8.59	11.15	15.98
+ IPS	19.67	24.38	31.08	9.12	12.01	16.75
+ LEVER + IPS	21.52	29.24	37.10	9.23	12.63	19.28
+ UDAPDR	24.71	34.02	44.93	-	-	-
+ SKIM (Ours)	**27.20**	**38.04**	**52.39**	**11.18**	**17.55**	**28.32**

lead to erroneous conclusions about the performance of a method. More results on the biased test sets are Tables 9 and 11 (Appendix).

5.3 Application to Sponsored Search

We demonstrate the large-scale real-world applicability of SKIM in sponsored search, where the task is to match user queries to a subset of relevant advertiser bid keywords, from potentially billions of keywords. This is a challenging problem which is further exacerbated by the systematic missing label bias in click logs, making it ideal for testing SKIM. To reliably evaluate SKIM, we do offline evaluation using a proprietary filter model and conduct online A/B tests on live search engine traffic.

Offline evaluation. Offline experiments are conducted on the proprietary LF-Query2Keyword-10M dataset, curated from clicks on a popular search engine. This dataset contains around 140M user queries and 10M ad keywords. The full landing page content corresponding to the ad keyword was treated as the unstructured metadata in SKIM. We compare Renée + SKIM with the deployed production model (Prod-Model), Renée [40] and Renée + LEVER [17]. We report $Hits@K$ where $K = 50, 100$, which is the number of pairs judged relevant among the top-K predictions by the proprietary filter model, averaged over all test queries. SKIM outperformed the closest baseline by 6 points in $Hits@100$ and 3 points in $Hits@50$. This goes on to show not just the effectiveness of SKIM, but also its large-scale real-world applicability for industry applications. More details about the proprietary filter model can be found in the Appendix F.

Online A/B test. For live deployment, SKIM was trained on a larger dataset similar to LF-Query2Keyword-10M with around 180M advertiser bid keywords and 170M user queries mined from click logs. A Renée + SKIM model was deployed on live traffic on a popular search engine to conduct an A/B study. SKIM increased the keyword density (number of predicted keywords which pass the various relevance and business filters in the sponsored search stack) by 16% indicating its ability to bring in good diverse keywords previously missed by an ensemble of state-of-the art dense retrievers, generative models, and XC algorithms. SKIM improved the impression-yield (avg. number of ads shown per query) by 1.02% and the click-yield (avg. number of ads clicked per query) by 1.23%. All improvements are statistically significant with p-value < 0.001.

Table 2: Offline evaluation results on proprietary query-keyword dataset LF-Query2Keyword-10M. SKIM outperforms LEVER by more than 6 absoulte points in $Hits@100$.

Method	**Hits@50**	**Hits@100**
Prod-Model	27.50	46.10
Renée	31.80	52.40
Renée + LEVER	31.20	52.20
Renée + SKIM	34.50	58.80

Table 3: Effect of providing meta-data to the SLM during the task-specific-distillation and large scale inference step. Providing meta-data compensates for lack of world-knowledge of the SLM and improves $R@100$ by 2.7 points on average.

		LF-Orcas-800K		LF-WikiHT-2M	
Model	**Metadata**	**R@25**	**R@100**	**R@25**	**R@100**
DEXML	✗	35.23	46.35	14.46	24.03
	✓	36.90	49.60	16.60	25.40
Renée	✗	34.54	48.80	13.58	24.29
	✓	38.00	52.40	17.60	28.30

5.4 Ablations

We conduct ablation studies to justify the design choices in SKIM. Specifically, we (i) demonstrate the importance of metadata in Step 1 of SKIM, (ii) show that retrieval augmentation (RA) [8] may be ineffective with biased training data, (iii) impact of SLM size on SKIM, and (iv) ineffectiveness of propensity-based methods. Further ablations on the the number of effect of new synthetic training pairs generated by SKIM, and the use of a pre-trained SLM for Step 1 of SKIM are provided in Appendix G.

Significance of unstructured metadata: We generate synthetic documents/queries using metadata and without metadata, and then compare the performance of the downstream trained XC models (see Table 3). The results reveal a significant gap in performance between the two scenarios, underscoring the critical role of metadata in recovering missing knowledge, even in unstructured form. Interestingly, without metadata, we see a marginal gain (compared to vanilla training on biased data) owing to *some* parametric world knowledge present in the SLM.

SKIM is complementary to retrieval-augmented XC: Recently, retrieval-augmentation (RA) works have improved performance in situations involving limited knowledge [8], that is, directly providing relevant knowledge as input to the retriever/generator improves performance. In this ablation (Table 4), we test what happens if one employs metadata augmentation, i.e., directly provide the relevant knowledge (metadata text) concatenated with the query text as input to the XC model. In a way, we are placing an upper bound on RA since we always provide the relevant information as

Table 4: Effect of Retrieval Aug. (RA) alongside SKIM on unbiased performance on LF-WikiTitlesHierarchy-2M dataset. For each configuration, we train a Renée model. When using RA, metadata is passed as input alongside the query during both train and test time.

Retrieval Aug.	SKIM	R@25	R@100
✗	✗	10.00	13.10
✗	✓	17.60	28.30
✓	✗	15.57	20.60
✓	✓	27.72	45.98

input alongside the query. Surprisingly, we observe that metadata augmented (or retrieval-augmented) XC model underperforms. This highlights an important point that one should de-bias the training dataset for RA to work. However, if we train using synthetic data generated by SKIM, we see a significant improvement (+17 points) in R@100. Interestingly, the XC model starts generalizing using the metadata provided as input and gives better performance as compared to the model trained using SKIM but without metadata-augmentation (compare rows 2 and 4 in Table 4). This shows that RA-like methods would be complementary to our approach and performance of RA might be limited if one does not de-bias the training dataset effectively. For more details see Appendix G.

To further show the complementary nature (refer Table 6 in Appendix), we use SKIM-augmented dataset with a recent retrieval augmented XC method OAK [69]. We again see the same trend.

Size of the SLM used in SKIM: Table 5, shows the effect of the size of SLM used in Step 1 of SKIM algorithm. We observe that even when the size of the SLM is decreased from 7B model to a 1.3B model the drop in R@100 is marginal drop ($< 1\%$). This makes SKIM extremely practical in limited compute.

Table 5: Effect of the size of the SLM used in SKIM's synthetic query generation step (i.e. Step 1) on the downstream unbiased performance of Renée + SKIM. We use LF-Orcas-800K dataset for this ablation. This is only a minor drop ($< 1\%$) when decreasing the size of the SLM from 7B to 1.3B parameters, making SKIM extremely practical even in low training compute regimes.

SLM	R@25	R@100
Phi-1.5 (1.3B)	37.28	51.66
Llama2 (7B)	38.00	52.4

Table 6: This ablation compares OAK and OAK + SKIM. We observe that SKIM provides complementary gains.

	LF-Orcas-800K		
Method	P@5	R@10	R@100
OAK	45.2	21.7	36.8
OAK + SKIM	49.7	26.6	49.1

Adding missing labels is more effective than accurate propensity estimates: We empirically show that directly adding diverse relevant missing labels directly in the training data is more effective than using propensity-based methods. For more details on this, refer to Appendix G.

6 Takeaways

Our findings motivate the following observations for practitioners working with real-world XC problems or large-scale retrieval in general.

1. More data is not more knowledge. In real world applications, more data is typically collected by increasing the time period in which clicks are collected using a deployed model. While this may lead to more query-document pairs, this doesn't guarantee additional knowledge in the newly collected dataset since the predictions of the deployed model is limited by the knowledge present in its training dataset. Thus, this cycle continues without adding any new beneficial knowledge, and consequently does not add new document labels representative of missing knowledge. As noted earlier (see Table 1, Figure 5 in the appendix), exposure to new knowledge is important to break this cycle, and SKIM takes the first step in this direction.

2. Building better offline evaluation. As discussed in Sec 5.2 relying on existing biased test sets for evaluation leads to misleading conclusions. Schultheis et al. [91] suggest the use of human labelled sets for reliable evaluation, but they are expensive to collect. However, with the advancements in LLMs, using them as reliable evaluation tools [80, 81, 92] can be a promising option. That being said, care should be taken while evaluating to not fit to the biases present in LLMs.

7 Conclusion

This paper studied a novel connection between the missing label issue in XC and the availability of world knowledge in the training data. The standard debiasing techniques such as propensity-scoring and imputation were found to be ineffective in mitigating knowledge-dependent systematic missing labels. To address this, it presented a novel approach, SKIM, a debiasing algorithm that addresses the missing knowledge problem at scale by using an SLM coupled with meta-data. Experimental results on multiple XC tasks showed strong improvements in Recall metrics when SKIM was applied to leading XC models. The real-world applicability of SKIM was demonstrated using an online A/B test on a popular search engine.

8 Acknowledgements

We would like to thank Neelabh Madan and Shikhar Mohan for their useful suggestions and feedback during paper writing. We would also like to thank the anonymous reviewers for their constructive feedback.

References

[1] Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. 2023. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540* (2023).

[2] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219* (2024).

[3] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[4] Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, et al. 2024. Nemotron-4 340B Technical Report. *arXiv preprint arXiv:2406.11704* (2024).

[5] Pranjal Aggarwal, Ameet Deshpande, and Karthik Narasimhan. 2023. SemSup-XC: semantic supervision for zero and few-shot extreme classification. In *Proceedings of the 40th International Conference on Machine Learning* (Honolulu, Hawaii, USA) *(ICML'23)*. JMLR.org, Article 11, 20 pages.

[6] Rahul Agrawal, Archit Gupta, Yashoteja Prabhu, and Manik Varma. 2013. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *Proceedings of the 22nd international conference on World Wide Web*. 13–24.

[7] Alon Albalak, Liangming Pan, Colin Raffel, and William Yang Wang. 2023. Efficient online data mixing for language model pre-training. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*.

[8] Akari Asai, Zexuan Zhong, Danqi Chen, Pang Wei Koh, Luke Zettlemoyer, Hannaneh Hajishirzi, and Wen-tau Yih. 2024. Reliable, adaptable, and attributable language models with retrieval. *arXiv preprint arXiv:2403.03187* (2024).

[9] Rohit Babbar and Bernhard Schölkopf. 2017. Dismec: Distributed sparse machines for extreme multi-label classification. In *Proceedings of the tenth ACM international conference on web search and data mining*. 721–729.

[10] Rohit Babbar and Bernhard Schölkopf. 2019. Data scarcity, robustness and extreme multi-label classification. *Machine Learning* 108, 8 (2019), 1329–1351.

[11] Elham J Barezi, Ian D Wood, Pascale Fung, and Hamid R Rabiee. 2019. A submodular feature-aware framework for label subset selection in extreme classification problems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 1009–1018.

[12] Jessa Bekker and Jesse Davis. 2018. Learning from positive and unlabeled data under the selected at random assumption. In *Second International Workshop on Learning with Imbalanced Domains: Theory and Applications*. PMLR, 8–22.

[13] Jessa Bekker and Jesse Davis. 2020. Learning from positive and unlabeled data: A survey. *Machine Learning* 109, 4 (2020), 719–760.

[14] K Bhatia, K Dahiya, H Jain, A Mittal, Y Prabhu, and M Varma. 2016. The extreme classification repository: Multi-label datasets and code. url: http://manikvarma.org/downloads/XC. *XMLRepository. html* 132 (2016).

[15] Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. 2015. Sparse local embeddings for extreme multi-label classification. *Advances in neural information processing systems* 28 (2015).

[16] Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. InPars: Data Augmentation for Information Retrieval using Large Language Models. arXiv:2202.05144 [cs.CL] https://arxiv.org/abs/2202.05144

[17] Anirudh Buvanesh, Rahul Chand, Jatin Prakash, Bhawna Paliwal, Mudit Dhawan, Neelabh Madan, Deepesh Hada, Vidit Jain, Sonu Mehta, Yashoteja Prabhu, Manish Gupta, Ramachandran Ramjee, and Manik Varma. 2024. Enhancing Tail Performance in Extreme Classifiers by Label Variance Reduction. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=6ARlSgun7J

[18] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems* 41, 3 (2023), 1–39.

[19] Andrew Collins, Dominika Tkaczyk, Akiko Aizawa, and Joeran Beel. 2018. A study of position bias in digital library recommender systems. *arXiv preprint arXiv:1802.06565* (2018).

[20] Nick Craswell, Daniel Campos, Bhaskar Mitra, Emine Yilmaz, and Bodo Billerbeck. 2020. Orcas: 18 million clicked query-document pairs for analyzing search. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2983–2989.

[21] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the TREC 2019 deep learning track. *arXiv preprint arXiv:2003.07820* (2020).

[22] Kunal Dahiya, Ananye Agarwal, Deepak Saini, Gururaj K, Jian Jiao, Amit Singh, Sumeet Agarwal, Purushottam Kar, and Manik Varma. 2021. SiameseXML: Siamese Networks meet Extreme Classifiers with 100M Labels. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 2330–2340. https://proceedings.mlr.press/v139/dahiya21a.html

[23] Kunal Dahiya, Nilesh Gupta, Deepak Saini, Akshay Soni, Yajun Wang, Kushal Dave, Jian Jiao, Gururaj K, Prasenjit Dey, Amit Singh, et al. 2023. Ngame: Negative mining-aware mini-batching for extreme classification. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 258–266.

[24] Kunal Dahiya, Deepak Saini, Anshul Mittal, Ankush Shaw, Kushal Dave, Akshay Soni, Himanshu Jain, Sumeet Agarwal, and Manik Varma. 2021. Deepxml: A deep extreme multi-label learning framework applied to short text documents. In *Proceedings of the 14th ACM international conference on web search and data mining*. 31–39.

[25] Kunal Dahiya, Sachin Yadav, Sushant Sondhi, Deepak Saini, Sonu Mehta, Jian Jiao, Sumeet Agarwal, Purushottam Kar, and Manik Varma. 2023. Deep encoders with auxiliary parameters for extreme classification. In *In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Long Beach, California.* https://www.microsoft.com/en-us/research/publication/deep-encoders-with-auxiliary-parameters-for-extreme-classification/

[26] K. Dahiya, S. Yadav, S. Sondhi, D. Saini, S. Mehta, J. Jiao, S. Agarwal, P. Kar, and M. Varma. 2023. Deep encoders with auxiliary parameters for extreme classification. In *KDD*.

[27] Zhuyun Dai, Vincent Y. Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B. Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot Dense Retrieval From 8 Examples. arXiv:2209.11755 [cs.CL] https://arxiv.org/abs/2209.11755

[28] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*. PMLR, 5547–5569.

[29] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander T Toshev, and Vaishaal Shankar. 2024. Data Filtering Networks. In *NeurIPS 2023 Workshop on Distribution Shifts: New Frontiers with Foundation Models*. https://openreview.net/forum?id=ZKtZ7KQ6G5

[30] Saumya Gandhi, Ritu Gala, Vijay Viswanathan, Tongshuang Wu, and Graham Neubig. 2024. Better Synthetic Data by Retrieving and Transforming Existing Datasets. *arXiv preprint arXiv:2404.14361* (2024).

[31] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks Are All You Need. arXiv:2306.11644 [cs.CL] https://arxiv.org/abs/2306.11644

[32] Chuan Guo, Ali Mousavi, Xiang Wu, Daniel N Holtmann-Rice, Satyen Kale, Sashank Reddi, and Sanjiv Kumar. 2019. Breaking the glass ceiling for embedding-based classifiers for large output spaces. *Advances in Neural Information Processing Systems* 32 (2019).

[33] Nilesh Gupta, Patrick Chen, Hsiang-Fu Yu, Cho-Jui Hsieh, and Inderjit Dhillon. 2022. ELIAS: End-to-End Learning to Index and Search in Large Output Spaces. *Advances in Neural Information Processing Systems* 35 (2022), 19798–19809.

[34] Nilesh Gupta, Fnu Devvrit, Ankit Singh Rawat, Srinadh Bhojanapalli, Prateek Jain, and Inderjit S Dhillon. 2024. Dual-Encoders for Extreme Multi-label Classification. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=dNe1T0Ahby

[35] Nilesh Gupta, Fnu Devvrit, Ankit Singh Rawat, Srinadh Bhojanapalli, Prateek Jain, and Inderjit S Dhillon. 2024. Efficacy of Dual-Encoders for Extreme Multi-label Classification. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=dNe1T0Ahby

[36] Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. *arXiv preprint arXiv:2203.09509* (2022).

[37] Qiyuan He, Yizhong Wang, and Wenya Wang. 2024. Can Language Models Act as Knowledge Bases at Scale? *arXiv preprint arXiv:2402.14273* (2024).

[38] Himanshu Jain, Venkatesh Balasubramanian, Bhanu Chunduri, and Manik Varma. 2019. Slice: Scalable linear extreme classifiers trained on 100 million labels for related searches. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 528–536.

[39] Himanshu Jain, Yashoteja Prabhu, and Manik Varma. 2016. Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 935–944.

[40] Vidit Jain, Jatin Prakash, Deepak Saini, Jian Jiao, Ramachandran Ramjee, and Manik Varma. 2023. Renee: End-to-end training of extreme classification models. In *Proceedings of Machine Learning and Systems*. To appear.

[41] Kristen Jaskie and Andreas Spanias. 2019. Positive and unlabeled learning algorithms and applications: A survey. In *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*. IEEE, 1–8.

[42] Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen

Eldan, Sivakanth Gopi, et al. 2023. Phi-2: The surprising power of small language models. *Microsoft Research Blog* (2023).

[43] Vitor Jeronymo, Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, Jakub Zavrel, and Rodrigo Nogueira. 2023. InPars-v2: Large Language Models as Efficient Dataset Generators for Information Retrieval. arXiv:2301.01820 [cs.IR] https://arxiv.org/abs/2301.01820

[44] Ting Jiang, Deqing Wang, Leilei Sun, Huayi Yang, Zhengyang Zhao, and Fuzhen Zhuang. 2021. Lightxml: Transformer with dynamic negative sampling for high-performance extreme multi-label text classification. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 7987–7994.

[45] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the tenth ACM international conference on web search and data mining*. 781–789.

[46] 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*. PMLR, 15696–15707.

[47] Masahiro Kato, Takeshi Teshima, and Junya Honda. 2019. Learning from positive and unlabeled data with a selection bias. In *International conference on learning representations*.

[48] Sujay Khandagale, Han Xiao, and Rohit Babbar. 2020. Bonsai: diverse and shallow trees for extreme multi-label classification. *Machine Learning* 109, 11 (2020), 2099–2119.

[49] Siddhant Kharbanda, Atmadeep Banerjee, Erik Schultheis, and Rohit Babbar. 2022. CascadeXML: Rethinking Transformers for End-to-end Multi-resolution Training in Extreme Multi-label Classification. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 2074–2087. https://proceedings.neurips.cc/paper_files/paper/2022/file/0e0157ce5ea15831072be4744cbd5334-Paper-Conference.pdf

[50] Siddhant Kharbanda, Devaansh Gupta, Erik Schultheis, Atmadeep Banerjee, Vikas Verma, and Rohit Babbar. 2024. Gandalf: Learning label correlations in Extreme Multi-label Classification via Label Features. https://openreview.net/forum?id=JuyFppXzh2

[51] Wonbin Kweon and Hwanjo Yu. 2024. Doubly Calibrated Estimator for Recommendation on Data Missing Not at Random. In *Proceedings of the ACM on Web Conference 2024* (Singapore, Singapore) *(WWW '24)*. Association for Computing Machinery, New York, NY, USA, 3810–3820. https://doi.org/10.1145/3589334.3645617

[52] Dong-Ho Lee, Jay Pujara, Mohit Sewak, Ryen White, and Sujay Jauhar. 2023. Making Large Language Models Better Data Creators. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 15349–15360.

[53] Jae-woong Lee, Seongmin Park, Mincheol Yoon, and Jongwuk Lee. 2023. uCTRL: Unbiased Contrastive Representation Learning via Alignment and Uniformity for Collaborative Filtering. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Taipei, Taiwan) *(SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 2456–2460. https://doi.org/10.1145/3539618.3592076

[54] Haoxuan Li, Yanghao Xiao, Chunyuan Zheng, Peng Wu, and Peng Cui. 2023. Propensity matters: measuring and enhancing balancing for recommendation. In *Proceedings of the 40th International Conference on Machine Learning* (Honolulu, Hawaii, USA) *(ICML'23)*. JMLR.org, Article 831, 13 pages.

[55] Haoxuan Li, Chunyuan Zheng, and Peng Wu. 2023. StableDR: Stabilized Doubly Robust Learning for Recommendation on Data Missing Not at Random. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=3VO1y5N7K1H

[56] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. 2024. DataComp-LM: In search of the next generation of training sets for language models. *arXiv preprint arXiv:2406.11794* (2024).

[57] Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic Data Generation with Large Language Models for Text Classification: Potential and Limitations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 10443–10461.

[58] Shuqi Lu, Di He, Chenyan Xiong, Guolin Ke, Waleed Malik, Zhicheng Dou, Paul Bennett, Tie-Yan Liu, and Arnold Overwijk. 2021. Less is More: Pretrain a Strong Siamese Encoder for Dense Text Retrieval Using a Weak Decoder. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2780–2791.

[59] Wenhao Lu, Jian Jiao, and Ruofei Zhang. 2020. Twinbert: Distilling knowledge to twin-structured compressed bert models for large-scale retrieval. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2645–2652.

[60] Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-Tuning LLaMA for Multi-Stage Text Retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 2421–2425. https://doi.org/10.1145/3626772.3657951

[61] Pratyush Maini, Skyler Seto, He Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. 2024. Rephrasing the web: A recipe for compute and data-efficient language modeling. *arXiv preprint arXiv:2401.16380* (2024).

[62] Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence* 42, 4 (2018), 824–836.

[63] Benjamin Marlin, Richard S Zemel, Sam Roweis, and Malcolm Slaney. 2012. Collaborative filtering and the missing at random assumption. *arXiv preprint arXiv:1206.5267* (2012).

[64] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 785–794.

[65] Tharun Kumar Reddy Medini, Qixuan Huang, Yiqiu Wang, Vijai Mohan, and Anshumali Shrivastava. 2019. Extreme Classification in Log Memory using Count-Min Sketch: A Case Study of Amazon Search with 50M Products. In *Advances in Neural Information Processing Systems*. 13244–13254.

[66] Anshul Mittal, Kunal Dahiya, Sheshansh Agrawal, Deepak Saini, Sumeet Agarwal, Purushottam Kar, and Manik Varma. 2021. DECAF: Deep Extreme Classification with Label Features. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* (Virtual Event, Israel) *(WSDM '21)*. Association for Computing Machinery, New York, NY, USA, 49–57. https://doi.org/10.1145/3437963.3441807

[67] Anshul Mittal, Shikhar Mohan, Deepak Saini, Suchith C. Prabhu, Jain jiao, Sumeet Agarwal, Soumen Chakrabarti, Purushottam Kar, and Manik Varma. 2024. Graph Regularized Encoder Training for Extreme Classification. arXiv:2402.18434 [cs.LG] https://arxiv.org/abs/2402.18434

[68] Anshul Mittal, Noveen Sachdeva, Sheshansh Agrawal, Sumeet Agarwal, Purushottam Kar, and Manik Varma. 2021. ECLARE: Extreme Classification with Label Graph Correlations. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 3721–3732. https://doi.org/10.1145/3442381.3449815

[69] Shikhar Mohan, Deepak Saini, Anshul Mittal, Sayak Ray Chowdhury, Bhawna Paliwal, Jian Jiao, Manish Gupta, and Manik Varma. 2024. OAK: Enriching Document Representations using Auxiliary Knowledge for Extreme Classification. In *Forty-first International Conference on Machine Learning*. https://openreview.net/forum?id=Cbacx90Wkt

[70] Akash Kumar Mohankumar, Bhargav Dodla, Gururaj K, and Amit Singh. 2023. Unified Generative & Dense Retrieval for Query Rewriting in Sponsored Search. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 4745–4751.

[71] M. Mohri, A. Rostamizadeh, and A. Talwalkar. 2012. *Foundations of Machine Learning*. MIT Press.

[72] Niklas Muennighoff, SU Hongjin, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. [n. d.]. Generative Representational Instruction Tuning. In *ICLR 2024 Workshop: How Far Are We From AGI*.

[73] Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. 2024. Scaling data-constrained language models. *Advances in Neural Information Processing Systems* 36 (2024).

[74] Yashoteja Prabhu, Anil Kag, Shilpa Gopinath, Kunal Dahiya, Shrutendra Harsola, Rahul Agrawal, and Manik Varma. 2018. Extreme multi-label learning with label features for warm-start tagging, ranking & recommendation. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 441–449.

[75] Yashoteja Prabhu, Anil Kag, Shrutendra Harsola, Rahul Agrawal, and Manik Varma. 2018. Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising. In *Proceedings of the 2018 World Wide Web Conference*. 993–1002.

[76] Yashoteja Prabhu and Manik Varma. 2014. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 263–272.

[77] Mohammadreza Qaraei, Erik Schultheis, Priyanshu Gupta, and Rohit Babbar. 2021. Convex Surrogates for Unbiased Loss Functions in Extreme Classification With Missing Labels. In *Proceedings of the Web Conference*. 3711–3720.

[78] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2020. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2010.08191* (2020).

[79] Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446* (2021).

[80] Hossein A Rahmani, Nick Craswell, Emine Yilmaz, Bhaskar Mitra, and Daniel Campos. 2024. Synthetic Test Collections for Retrieval Evaluation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2647–2651.

[81] Hossein A Rahmani, Clemencia Siro, Mohammad Aliannejadi, Nick Craswell, Charles LA Clarke, Guglielmo Faggioli, Bhaskar Mitra, Paul Thomas, and Emine Yilmaz. 2024. LLM4Eval: Large Language Model for Evaluation in IR. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3040–3043.

[82] Ankit Singh Rawat, Aditya K Menon, Wittawat Jitkrittum, Sadeep Jayasumana, Felix Yu, Sashank Reddi, and Sanjiv Kumar. 2021. Disentangling sampling and labeling bias for learning in large-output spaces. In *International conference on machine learning*. PMLR, 8890–8901.

[83] Sashank J Reddi, Satyen Kale, Felix Yu, Daniel Holtmann-Rice, Jiecao Chen, and Sanjiv Kumar. 2019. Stochastic negative mining for learning with large output spaces. In *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 1940–1949.

[84] Andy Rosenbaum, Saleh Soltan, Wael Hamza, Marco Damonte, Isabel Groves, and Amir Saffari. 2022. CLASP: Few-Shot Cross-Lingual Data Augmentation for Semantic Parsing. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 444–462.

[85] Andy Rosenbaum, Saleh Soltan, Wael Hamza, Yannick Versley, and Markus Boese. 2022. LINGUIST: Language Model Instruction Tuning to Generate Annotated Utterances for Intent Classification and Slot Tagging. In *Proceedings of the 29th International Conference on Computational Linguistics*. 218–241.

[86] Jon Saad-Falcon, Omar Khattab, Keshav Santhanam, Radu Florian, Martin Franz, Salim Roukos, Avirup Sil, Md Arafat Sultan, and Christopher Potts. 2023. UDAPDR: Unsupervised Domain Adaptation via LLM Prompting and Distillation of Rerankers. arXiv:2303.00807 [cs.IR]

[87] Devendra Singh Sachan, Mike Lewis, Dani Yogatama, Luke Zettlemoyer, Joelle Pineau, and Manzil Zaheer. 2023. Questions are all you need to train a dense passage retriever. *Transactions of the Association for Computational Linguistics* 11 (2023), 600–616.

[88] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased recommender learning from missing-not-at-random implicit feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 501–509.

[89] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: debiasing learning and evaluation. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*. 1670–1679.

[90] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: debiasing learning and evaluation *(ICML'16)*. JMLR.org, 1670–1679.

[91] Erik Schultheis, Marek Wydmuch, Rohit Babbar, and Krzysztof Dembczynski. 2022. On missing labels, long-tails and propensities in extreme multi-label classification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1547–1557.

[92] Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2024. Large language models can accurately predict searcher preferences. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1930–1940.

[93] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[94] Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query Expansion with Large Language Models. *arXiv preprint arXiv:2303.07678* (2023).

[95] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2019. Doubly robust joint learning for recommendation on data missing not at random. In *International Conference on Machine Learning*. PMLR, 6638–6647.

[96] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2021. Combating Selection Biases in Recommender Systems with a Few Unbiased Ratings. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* (Virtual Event, Israel) *(WSDM '21)*. Association for Computing Machinery, New York, NY, USA, 427–435. https://doi.org/10.1145/3437963.3441799

[97] Tong Wei, Wei-Wei Tu, Yu-Feng Li, and Guo-Ping Yang. 2021. Towards robust prediction on tail labels. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1812–1820.

[98] Chenxi Whitehouse, Monojit Choudhury, and Alham Aji. 2023. LLM-powered Data Augmentation for Enhanced Cross-lingual Performance. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 671–686.

[99] Marek Wydmuch, Kalina Jasinska-Kobus, Rohit Babbar, and Krzysztof Dembczynski. 2021. Propensity-scored probabilistic label trees. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2252–2256.

[100] Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2024. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems* 36 (2024).

[101] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808* (2020).

[102] Hui Ye, Zhiyu Chen, Da-Han Wang, and Brian Davison. 2020. Pretrained generalized autoregressive model with adaptive probabilistic label clusters for extreme multi-label text classification. In *International Conference on Machine Learning*. PMLR, 10809–10819.

[103] Jing Liu Kai Liu Ruiyang Ren Wayne Xin Zhao Daxiang Dong Hua Wu Yingqi Qu, Yuchen Ding and Haifeng Wang. 2021. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering. In *In Proceedings of NAACL*.

[104] Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2019. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. *Advances in neural information processing systems* 32 (2019).

[105] Jiong Zhang, Wei-Cheng Chang, Hsiang-Fu Yu, and Inderjit Dhillon. 2021. Fast multi-resolution transformer fine-tuning for extreme multi-label text classification. *Advances in Neural Information Processing Systems* 34 (2021), 7267–7280.

[106] Wenhao Zhang, Wentian Bao, Xiao-Yang Liu, Keping Yang, Quan Lin, Hong Wen, and Ramin Ramezani. 2020. Large-scale Causal Approaches to Debiasing Post-click Conversion Rate Estimation with Multi-task Learning. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) *(WWW '20)*. Association for Computing Machinery, New York, NY, USA, 2775–2781. https://doi.org/10.1145/3366423.3380037

A Theoretical Proofs

Lemma 1. *If $y_{il} = 0 \quad \forall(\mathbf{x}_i, \mathbf{z}_l) \in D_m$, then for any test pair $(\mathbf{x}, \mathbf{z}) \sim D_m$, $R(\mathbf{x}, \mathbf{z}, k_m) \perp\!\!\!\perp \mathcal{D}$ where $\mathcal{D} = \{\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N, \{\mathbf{z}_l\}_{l=1}^L\}$ is the training dataset.*

Proof. As $R(\mathbf{x}, \mathbf{z}, k_{\mathbf{x},\mathbf{z}})$ is a deterministic function of its parameters, the following conditional independence holds true:

$$\{\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N, \{\mathbf{z}_l\}_{l=1}^L\} \perp\!\!\!\perp \mathbf{x}, \mathbf{z}, k_m | (\mathcal{K} \setminus k_m) \tag{1}$$

The independence w.r.t $\mathbf{x}, \mathbf{z}$ is due to sampling independence. The independence w.r.t k_m arises because, as since $y_{il} = 0 \quad \forall(\mathbf{x}_i, \mathbf{z}_l) \in D_m$, the rest of the documents in $\mathcal{D}$ depend only on $\mathcal{K} \setminus k_m$ owing to Postulates 1 and 3.

Additionally, the following also holds true owing to Postulate 2 and independent sampling:

$$\mathbf{x}, \mathbf{z}, k_m \perp\!\!\!\perp (\mathcal{K} \setminus k_m) \tag{2}$$

By combining (1) and (2) through the contraction lemma of independence, and then using the determinism of $R(\mathbf{x}, \mathbf{z}, k_{\mathbf{x},\mathbf{z}})$ we get the desired:

$$R(\mathbf{x}, \mathbf{z}, k_m) \perp\!\!\!\perp \{\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N, \{\mathbf{z}_l\}_{l=1}^L\}. \tag{3}$$

□

Corollary 2. *If $y_{il} = 0 \quad \forall(\mathbf{x}_i, \mathbf{z}_l) \in D_m$, then for any test pair $(\mathbf{x}, \mathbf{z}) \sim D_m$, $R(\mathbf{x}, \mathbf{z}, k_m) \perp\!\!\!\perp \mathcal{M}$; where $\mathcal{M}$ is a predictive model trained deterministically from $\{\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N, \{\mathbf{z}_l\}_{l=1}^L\}$.*

Proof. As $\mathcal{M}$ is a deterministic function of $\{\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N, \{\mathbf{z}_l\}_{l=1}^L\}$, the following holds true:

$$\mathcal{M} \perp\!\!\!\perp \mathbf{x}, \mathbf{z}, k_m | \{\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N, \{\mathbf{z}_l\}_{l=1}^L\} \tag{4}$$

By combining (4) with (3) through the contraction lemma, we get the desired result. □

Theorem 3. *Let n be the number of clicked query-doc pair sampled for the training data, B be the exposure bias while sampling, and $\bar{F}_m, p_m$ be defined as above. Then, for any $0 \leq \delta \leq 1$, with probability at least $1 - \delta$, at least $\max_{m=1}^K \frac{\bar{F}_m}{B} e^{-B.p_m.n} - \frac{1}{2\sqrt{2n}}(\log \frac{2K}{\delta}^{\frac{3}{2}})$ fraction of the true relevant label distribution will correspond to systematic missing label region and therefore will be irrecoverable by supervised training.*

Proof. Marginal probability of not sampling a knowledge item at all during training data generation is $(1 - B.p_m)^n$. The associated probability of systematic missing labels then is $p_m(1 - B.p_m)^n$

Now, the expected total loss due to all lost knowledge items, $\bar{E}$ is:

$$\bar{E} = \frac{1}{B}\mathbb{E}\left[\sum_{m=1}^{K} p_m(1 - B.p_m)^n\right] \tag{5}$$

$$\geq \frac{1}{B}\mathbb{E}\left[\sum_{m=m'}^{K} p_m(1 - B.p_m)^n\right] \tag{6}$$

$$\geq \frac{1}{B}\max_{m=m'}^{K} \bar{F}_{m'}(1 - B.p_{m'})^n \tag{7}$$

$$\approx \frac{1}{B}\max_{m=m'}^{K} \bar{F}_{m'} e^{-B.p'_m.n} \tag{8}$$

where the last step uses the approximation that can be accurate when the value of $p_{m'}$ is small, *i.e.* on the long-tail.

Next, by using Extended McDiarmid's inequality [71], the value E which is the probability of systematic missing label error for any random sample of dataset can be bounded as follows:

$$\mathbb{P}(E - \bar{E} \leq \epsilon) \leq 2q + 2\exp\left\{-2\frac{(\epsilon - q.np)^2}{np^2}\right\} \tag{9}$$

where for some m', $p = p_{m'}$ and

$$q = \sum_{m=1}^{m'} (1 - p_m)^n \tag{10}$$

$$\leq m'(1 - p_{m'})^n \tag{11}$$

Plugging these back in Extended McDiarmid's inequality:

$$\mathbb{P}(E - \bar{E} \leq \epsilon) \leq \max_{m'=1}^{K} 2m'(1 - p_{m'})^n + 2\exp\left\{-2\frac{(\epsilon - q.np_{m'})^2}{np_{m'}^2}\right\} \tag{12}$$

$$\leq \max_{p} 2K(1-p)^n + 2\exp\left\{-2\frac{(\epsilon - q.np)^2}{np^2}\right\} \tag{13}$$

$$= \max_{p} 2Ke^{n\log(1-p)} + 2e^{\left\{-2\frac{(\epsilon - q.np)^2}{np^2}\right\}} \tag{14}$$

$$\leq \max_{p} 2Ke^{-np} + 2e^{\left\{-2\frac{\epsilon^2}{np^2}\right\}} \tag{15}$$

$$\leq 2Ke^{-2n^{\frac{1}{3}}\epsilon^{\frac{2}{3}}} \tag{16}$$

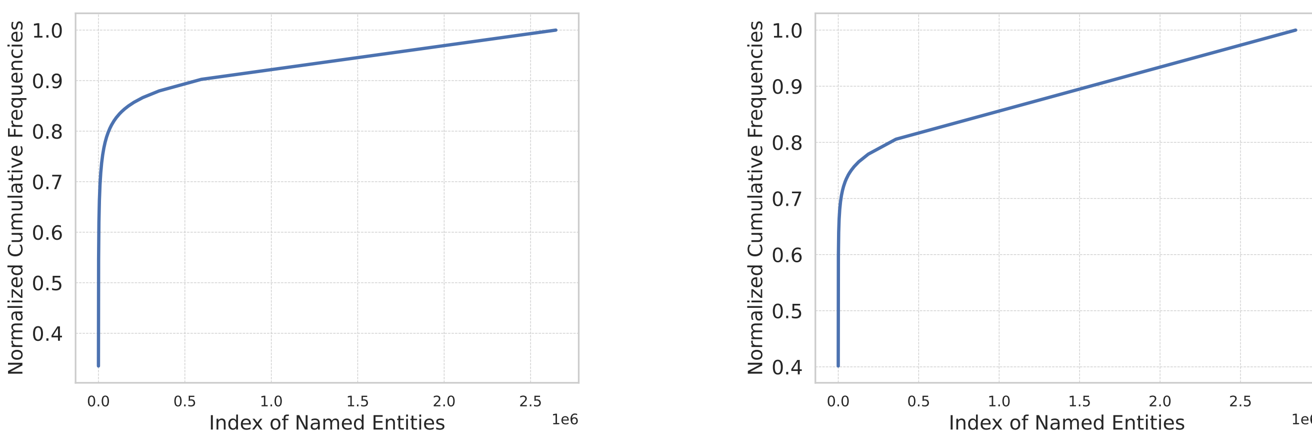
As a result,

$$\mathbb{P}(E - \bar{E} \leq \epsilon) \leq 2Ke^{-n^{\frac{1}{3}}\epsilon^{\frac{2}{3}}} \tag{17}$$

In other words, for any small $\delta \leq 1$, with probability at least $1 - \delta$ on sampling the dataset $\mathcal{D}$, the error E of systematic missing labels is at least: $max_{m=1}^{K} \frac{\bar{F}_m}{B} e^{-B.p_m.n} - \frac{1}{2\sqrt{2n}}(\log \frac{2K}{\delta}^{\frac{3}{2}})$.

□

B Long Tail of Knowledge required in XC Applications



(a) LF-Orcas-800K **(b) LF-WikiTitlesHierarchy-2M**

Figure 3: We use named entity as a proxy for knowledge to show that XC applications require vast and long-tail knowledge. On the X-axis we plot the index of the named entities and on the Y-axis we show the normalized cumulative frequency of the named entities. In LF-Orcas-800K dataset [26], where the task is matching user queries to web-page URL, around 80% of the knowledge is covered by 2.23% entities. While in the LF-WikiTitlesHierarchy-2M dataset where the task is to match Wikipedia titles to categories, around 11.28% of the entities cover the same fraction of knowledge. This highlights the extensive and long-tail knowledge required by XC applications.

C Dataset Curation and Statistics

This section describes the train-test splits used for training and evaluating different XC models. We benchmark on two public large-scale short-text XC datasets, namely LF-Orcas-800K and LF-WikiTitlesHierarchy-2M. We also evaluate SKIM on the proprietary query to ad-keyword dataset curated from clicks logs of a popular search engine, we call it LF-Query2Keyword-10M. Below we outline the creation procedure and statistics for each dataset.

LF-Orcas-800K: We use the LF-Orcas-800K dataset released by Dahiya et al. [25] for XC, and originally proposed by Craswell et al. [20]. The LF-Orcas-800K dataset models the task of mapping short user queries to web-page URLs. For example, a user query like "*exons definition biology*" may be relevant to URLs like "*www.merriam-webster.com dictionary/exon*" and "*en.wikipedia.org/wiki/Exon*". For unbiased evaluation we curate a small unbiased test dataset using the human labelled test sets available on TREC-19 and TREC-20 Deep Learning competitions [21]. Statistics of the dataset are provided in Table 7.

LF-WikiTitlesHierarchy-2M: This dataset was created by mining the Wikipedia dump [3]. From the dump, we extract all Wikipedia titles and link each title to its associated categories and parent categories, as described in [17]. We assume this dataset to be relatively unbiased since it was annotated by humans. To simulate click bias (controlled bias), we only include query-document pairs that appear in the top-200 predictions of a pre-trained `msmarco` retriever in the training set (refer Algorithm 1). This introduced bias mimics practical scenarios where users are shown a limited number of documents (or items) for a given query. Consequently, if an item is not in the top shortlist retrieved by the model, it will never be clicked and thus will not appear in the dataset curated from click logs. For unbiased evaluation, we use a test set that includes the complete ground truth without any simulation bias. Statistics of the dataset are provided in Table 7.

Table 7: Dataset statistics for train, biased test set, and unbiased test set. Due to lack of space, average points per document and average documents per point have been abbrevaiated as Av. PPL and Av. LPP respectively. LF-WikiTitlesHierarchy-2M has been shown as LF-WikiHT-2M. LF-Query2Keyword-10M is abbrevaiated as LF-Q2K-10M. Some details about the proprietary datasets has been redacted by "-". We only provide approximate values for number of documents, train queries and test queries for LF-Q2K-10M.

Dataset	# Documents	Train Set			Biased Test Set			Unbiased Test Set		
		# Data Pts	Av. PPL	Av. LPP	# Data Pts	Av. PPL	Av. LPP	# Data Pts	Av. PPL	Av. LPP
LF-Orcas-800K	797322	7360881	16.133	1.747	2547702	5.676	1.776	88	1.009	27.750
LF-WikiHT-2M	1966221	5418305	5.259	1.908	1355377	1.317	1.911	1355377	35.044	50.838
LF-Q2K-10M	~10000000	~140000000	-	-	~16000000	-	-	~16000000	-	-

Algorithm 1: Simulated bias dataset creation process for LF-WikiTitlesHierarchy-2M dataset.

Input: Human annotated (*unbiased*) dataset $\mathcal{D}$, Pre-trained `msmarco` model $\mathcal{M}$, Parameter k controlling the degree of exposure
Ouput: Biased Dataset $\mathcal{D}_{biased}$
for query q in $\mathcal{D}$ **do**
  $\mathcal{S}_q \longleftarrow$: Set of relevant documents for query q in $\mathcal{D}$.
  $\mathcal{M}_q \longleftarrow$: Set of `top-k` predictions for query q by model $\mathcal{M}$.
  $\mathcal{T}_q \longleftarrow$: $\mathcal{S}_q \cap \mathcal{M}_q$
  for doc d *in* $\mathcal{T}_q$ **do**
    $\mathcal{D}_{biased} \longleftarrow \mathcal{D}_{biased} \cup \{q, d\}$

D Baseline Implementation Details

This section describes the details of the baselines used in the main paper. We first describe both base XC methods, namely Renée and DEXML, and then move on to the baselines against which we compare SKIM.

D.1 Base XC Methods

Renée [40]: This is a recent XC approach that makes use of one-versus-all (OvA) to represent each document. Query representations are derived from a 6-layer DistilBERT [4] encoder. We use the source code provided here for training Renée.
DEXML [35]: This is a state-of-the-art dual encoder approach that derives the representations of both queries and documents using a 6-layer DistilBERT encoder. We use the provided source code here for training DEXML. Due to the larger size of our datasets (800K documents in LF-Orcas-800K and 2M documents in LF-WikiTitlesHierarchy-2M), the hard negative mining configuration provided here is used in all experiments.

D.2 Baselines

We compare SKIM with augmentation, propensity based and SLM-based rewriting approaches. Details of the baselines are provided below.

[3]https://dumps.wikimedia.org/enwiki/20240420/
[4]https://huggingface.co/docs/transformers/en/model_doc/distilbert

Gandalf [50]: This is a graph-based data augmentation method that effectively models document-document correlations. As described in [50], we use a threshold of 0.1 to create the document-document augmentation matrix.
LEVER [17]: This is a recent approach that improves the tail performance of any XC classifier by augmenting the ground truth using soft scores derived from a Siamese model. We use the source code provided here for training LEVER.
Inverse Propensity Scoring (IPS) [39, 77]: For propensity-based training, we derive per-document propensity estimates using the propensity model described in [39]. For training Renée, we reweigh the positives using the factor described in [77]. In the case of DEXML, we simply use the propensity scores from [39] to reweigh the positives.
LEVER + IPS: Drawing inspiration from the success of doubly robust methods [54] that combine augmentation and propensity based methods, we create a baseline that combines reweighs the positives using IPS and augments the ground truth using LEVER.
UDAPDR: We used a slightly modified version of UDAPDR [86] as a baseline. Specifically, we used a finetuned SLM on our generated GPT4 demonstrations but without metadata. This is unlike [43, 86], where the authors directly prompted a pretrained SLM for document/query generation. In fact, we found that directly prompting a pretrained SLM leads to drastically inferior document generation (see Figure 6). However, as done in [43, 86], we do not employ a pretrained/finetuned re-ranker to filter the generated query-document pairs. We train the retriever on all the generated query-document pairs. We provide the generated query-document pairs used in our experiments.

E SKIM Implementation Details

This section discusses implementation details, hyperparameters used and the hardware employed for SKIM. We cover: (i) task specific distillation of the SLM from LLM, (ii) Large-scale inference to generate synthetic queries (Step 1), and finally (iii) mapping synthetic queries to actual queries (Step 2) of SKIM.

(i) Task-specific Distillation: To create the distillation / finetuning data for the SLM, we use GPT4 as the LLM. We use the prompts mentioned in section E.1. We use the fast OpenAI access key to generate close to ~40K-50K responses from GPT4. This takes a few hours to collect. Next, we use Low Rank Adaptation (LoRA) technique to finetune SLMs on GPT4 responses. We use the implementation provided here: litgpt for finetuning SLM using LoRA. We finetune the SLM for a total of 2 epochs over the GPT4 responses for all datasets, using an effective batch-size of 64 and `bfloat16` weights, on H100 80GB GPU. We train `meta-llama/Llama-2-7b-hf` (Llama2 7B) for our main experiments, and train `microsoft/phi-1_5` (Phi1.5 1.3B) for ablation.

(ii) Large-scale inference to generate synthetic queries: Once we have the distilled / finetuned SLM, we run inference on all the (document, unstructured metadata) pairs present in our dataset to generate synthethic queries. For LF-Orcas-800K, we cap the tokens in the metadata to be 750 tokens, and limited the generated text to be at max 250 tokens. In LF-WikiTitlesHierarchy-2M, we cap the metadata to 1000, and the generations to 1000 tokens. We run large-scale inference of SLM to generate synthetic queries (Step 1 of SKIM) again on RTX A6000 GPUs.

(iii) Mapping synthetic queries to actual queries (Step 2): Once we have synthetic queries, we map them onto the actual queries using ANNS. We use the popular implementation of Malkov and Yashunin [62] provided here: hnswlib. This takes a few minutes to calculate nearest neighbours on 10M document XC dataset on 96 core CPU machine without any GPUs. We linearly scale the normalized cosine score in $[0, 1]$, and then select the similarity threshold τ to filter the actual queries we add to the new training set for a document. For all datasets, we set default as $\tau = 0.8$.

E.1 Prompts for Synthetic Query (Document) Generation

We prompt GPT-4 and, subsequently, the fine-tuned LLAMA-2 on two datasets: LF-WikiTitlesHierarchy-2M and LF-Orcas-800K. For both datasets, we incorporate available metadata into the prompts. The GPT-4 prompts include a description of the task and manually curated in-context examples to ensure the generation of high-quality synthetic queries (or documents). The GPT-4 responses are then used to fine-tune the SLM. Once fine-tuned, the SLM inference is run by providing only the query (or document) and associated metadata, as through fine-tuning, the SLM has absorbed the task-specific information. The shorter SLM prompt also aids in scaling the to large-scale XC datasets.

E.2 Time and Hardware requirements of SKIM

Table 8: Time taken and Hardware used in the different steps of SKIM algorithm. In brackets, we mentioned the number of RTX A6000 48GB GPUs used for large-scale SLM inference. For SLM finetuning, we used H100 80GB GPU.

	Generating Examples from LLM (hrs)	Finetuning SLM (hrs)	SLM Inference (hrs)
LF-Orcas-800K	10	3	0.16 (128)
LF-WikiHT-2M	12	4	43.40 (128)

F Sponsored Search

In this section we provide more details about the metadata and the proprietary filter model used in our sponsored search experiments.

Metadata used. For every ad-keyword, we use the advertiser provided full landing page as unstructured metadata.

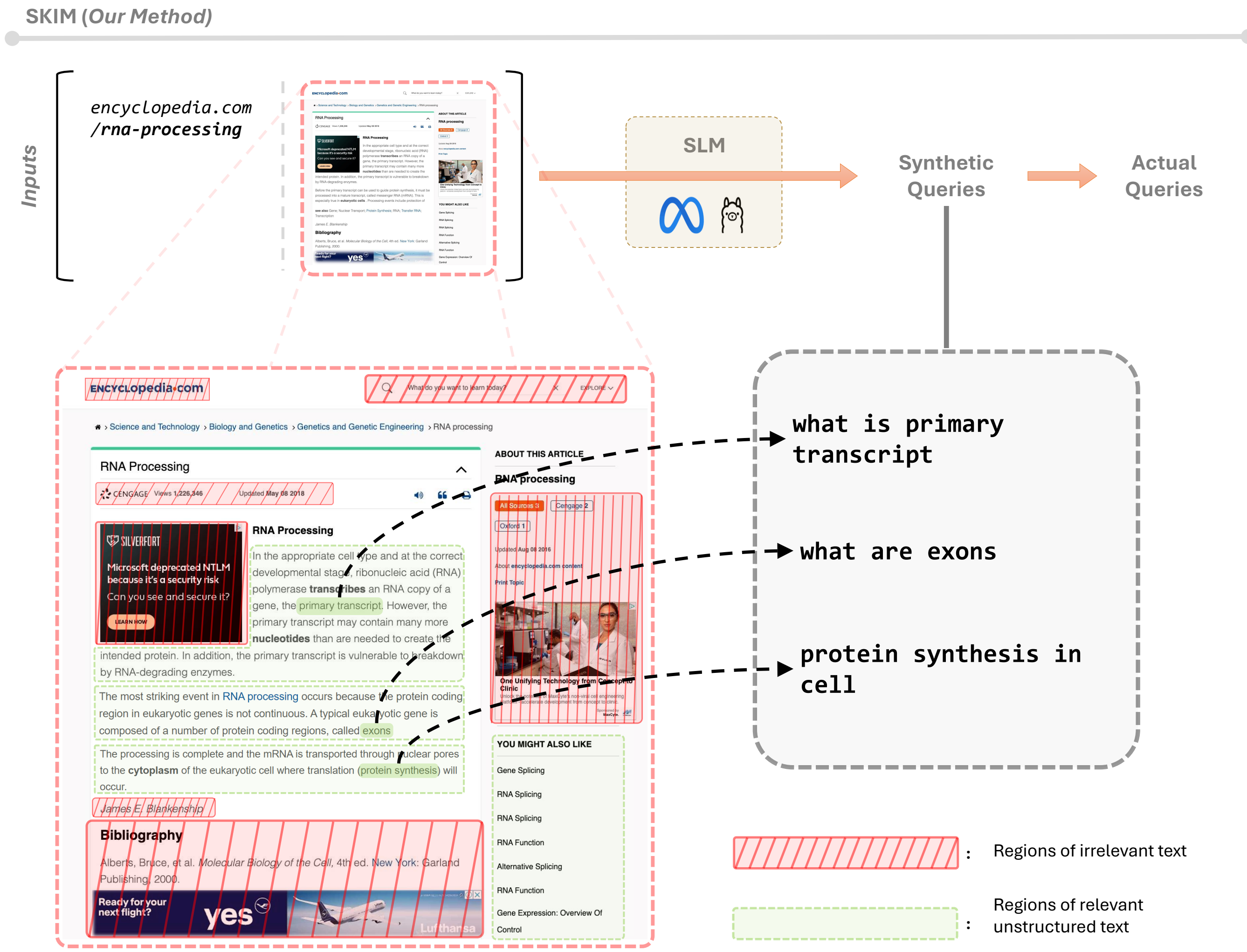


Figure 4: Intuitive visual explanation of how a finetuned SLM uses implicit knowledge contained in unstructured metadata to generate diverse synthetic queries that are representative of different knowledge concepts that could be absent in the training dataset. In other words, the finetuned SLM is able to *skim* through the unstructured metadata, ignoring the non-relevant text, and generate only the relevant synthetic queries that are representative of different knowledge concepts about the document.

Proprietary Filter Model. This model measures the relevance between a given query and an ad-keyword pair. This is implemented as a cross-encoder, initialized from BERT-Large, trained on large-scale human labelled (query, ad-keyword) pairs.

LF-WikiTitlesHierarchy-2M GPT-4 Synthetic Query Generation Prompt.

Task Instructions **/* Describing the task */**
You are given a Wikipedia article, specifically the title and body text of the article. Your job is to generate immediate Wikipedia categories as well as the Wikipedia categories for the generated immediate Wikipedia categories using the information provided in the body text of the article.
When generating Wikipedia categories, make sure to follow below mentioned guidelines for a valid Wikipedia category:
1. Relevance: The categories you generate should be directly relevant to the topic of the Title. They should describe key aspects of the subject matter. For example, if given an Title about dogs, relevant categories might include "Mammals," "Pets," "Animal Behavior," or "Dog Breeds." Categories like "Astronomy" or "Cooking" would not be relevant to a dog-related Title.
2. Specificity: Wikipedia's category system is organized hierarchically, with broader categories containing more specific subcategories. Try to generate categories that aim to place the Title in the most specific category that applies.
First generate the immediate Wikipedia categories which are directly relevant to the article. After that, generate Wikipedia categories for the earlier generated immediate Wikipedia categories. Make sure to follow the guidelines that are mentioned above.
Below are some examples that should provide more clarity about the task.
Example 1: **/* In-Context Example */**
Wikipedia Title
Sellankandal **/* Input query */**
Wikipedia article begins **/* Meta-Data */**
Sellankandal is a village situated 10 km inland from coastal Puttalam city in the North Western Province of Sri Lanka. It is the primary settlement of people of Black African descent in Sri Lanka called Kaffirs who until the 1930s spoke a Creole version of Portuguese. Most villages speak Sinhala and are found throughout the country as well as in the Middle East. The Baila type of music, very popular in Sri Lanka since the 1980s, originated centuries ago among this 'kaffir' community.
Wikipedia article ends

Task Output **/* Synthetic Generated Documents */**
Immediate Wikipedia Categories
Populated places in North Western Province, Sri Lanka
Populated places in Puttalam District
African diaspora in Sri Lanka

Wikipedia Categories for Immediate Wikipedia Categories **/* Example Category Hierarchy for this Query */**
Populated places in North Western Province, Sri Lanka
  Populated places in Sri Lanka by province
  Geography of North Western Province, Sri Lanka
Populated places in Puttalam District
  Populated places in North Western Province, Sri Lanka
  Populated places in Sri Lanka by district
  Geography of Puttalam District
African diaspora in Sri Lanka
  African diaspora in Asia
  Ethnic groups in Sri Lanka

Now perform the task for the following:
Please do not use any external knowledge to generate Wikipedia categories, the information must come only from the provided article, do not generate anything extra that is not there in the article. **Make sure that the categories generated are actual Wikipedia Categories.** Generate as much relevant immediate Wikipedia categories as possible.
Wikipedia Title
{title}
Wikipedia article begins
{content}
Wikipedia article ends
Task Output

LF-Orcas-800K GPT-4 Synthetic Query Generation Prompt.

Task Instruction **/* Describing the task */**
You are given a URL, along with some relevant search queries for this URL. Note that each of these relevant search queries is a "good" match for the given URL. Additionally, you are given some metadata in the form of the webpage text of this URL. You may use the relevant search queries and the webpage text to understand how the webpage text for this URL contains information that can answer or partially answer the relevant user queries. You need to solve the following task for this given URL. Please use only the information provided in the form of webpage text for the given URL, and the relevant search queries to solve the Task.
Query Generation from Webtext Task
Generate targeted search queries for the given URL using the webtext. The generated search queries contain tokens different from tokens of the URL. Generate upto 10 relevant search queries. Only generate new search queries, do not copy anything given in the relevant queries section.
Following are some examples for the task: **/* In-Context Example */**
URL: https://www.laserspineinstitute.com/back_problems/vertebrae/l3/ **/* Input Document */**
Relevant queries **/* Associated Queries in Training Data */**
l3 compression fracture
l3-l4 symptoms
l3 and l4 vertebrae
l3 nerve root compression

Webpage text for URL begins **/* Meta-Data */**
Vertebrae Injury Vertebrae Fracture Back Vertebrae Vertebrae Pain Vertebrae Column Spinal Cord Vertebrae Between the Vertebrae Lumbar Spine Vertebrae Vertebrae Treatment Vertebrae Compression Vertebrae Compression Fracture Vertebrae Disc Vertebrae Nerve Spinal Column Vertebrae Vertebrae Surgery Neck Vertebrae Spine Vertebrae Spinal Vertebrae Compressed Vertebrae The L3 vertebra is located in the lumbar spine, which is in the lower back portion of the spinal column. The lumbar spine typically has five vertebrae, though some people range from four to six vertebrae in the lumbar region. The purpose of the lumbar spine is to stabilize and support the weight of the body, while still allowing the spine to move and bend freely. Because of the versatile nature of the lumbar spine, the vertebrae in this area are prone to injury and the development of spine conditions. The L3 vertebra is particularly susceptible to injury because it is the middle vertebra in the lumbar spine, which means it handles the most stress when the lumbar twists and bends. The L3 vertebra holds most of the weight and stress of the body compared to the other vertebrae in the lumbar spine. Because of this, there are several spine conditions that can develop at the L3 vertebra and impact the surrounding nerve root, disc and/or joint. The most common spine conditions at the L3 vertebra include: Herniated disc Bulging disc Bone spurs Spondylosis Spondylolisthesis Arthritis of the spine There are several other conditions that may develop as a result of vertebral compression in the lumbar spine, which often impacts the disc and joints in the spine. If a spine condition occurs in the L3 vertebra, the symptoms will likely include chronic lower back pain and radiating pain in the buttock and leg of the impacted side. Additionally, the leg and foot might feel weak and numb due to the impacted nerve root being unable to send strong signals to the extremities.
Webpage text for URL ends
Output
Query Generation from Webtext Task **/* Useful Queries extracted from this web-page */**
location of L3
how many vertebrae in lumbar spine
causes for spondylosis
role of lumbar spine
chronic back pain
weak and numb feet

Now perform the Query Generation from Webtext task. Ensure that the generated query can be answered or contains information relevant to the webtext of the given URL. Try to generate new kinds of queries that do not overlap with the given relevant queries. Generate at most 10 search queries for the Query Generation from Webtext Task. Do not generate any query that has a text overlap with either the URL or the given relevant queries
URL {url}
Relevant queries
{relevant queries}
Webpage text for URL begins
{doc title}
{doc body}
Webpage text for URL ends
Output

Table 9: Comparison of SKIM with baseline methods on the LF-Orcas-800K dataset. The first two sets of rows present evaluations on the unbiased test set, a small-scale human-annotated dataset derived from the TREC-19 and TREC-20 Deep Learning competitions [21]. The following two sets of rows show results on the biased test set from [26]. nDCG@k has been abbreviated as N@k for the sake of brevity. PSP metrics on the biased test set are reported in Table 10.

Unbiased Metrics												
Method	**P@1**	**P@3**	**P@5**	**N@1**	**N@3**	**N@5**	**R@1**	**R@3**	**R@5**	**R@10**	**R@25**	**R@100**
DEXML	65.91	50.00	42.05	65.91	54.30	48.90	6.42	12.47	15.96	21.77	28.32	39.08
+ Gandalf	68.18	53.79	44.09	68.18	57.70	51.20	6.86	13.99	17.10	22.39	29.87	40.52
+ LEVER	70.45	54.17	43.64	70.45	58.65	51.39	6.61	13.84	16.51	21.70	29.26	40.45
+ IPS	70.45	54.55	45.45	70.45	58.91	52.87	7.08	13.43	17.50	22.31	29.14	39.42
+ LEVER + IPS	64.77	56.06	46.14	64.77	58.92	52.62	6.64	14.17	18.00	23.47	30.92	41.04
+ UDAPDR	68.18	52.65	46.36	68.18	57.40	53.33	6.51	12.89	18.72	26.42	33.71	47.70
+ SKIM (Ours)	65.91	58.33	50.68	65.91	60.98	56.35	7.07	14.43	18.78	26.99	36.90	49.60
Renée	65.91	52.65	45.68	65.91	56.71	52.26	6.64	13.86	17.70	21.75	28.22	38.42
+ Gandalf	70.45	54.92	44.09	70.45	59.41	52.18	7.63	14.33	17.22	20.89	27.58	34.66
+ LEVER	68.18	54.17	45.45	68.18	58.34	52.62	7.47	14.10	17.24	22.58	28.40	39.52
+ IPS	65.91	51.89	40.91	65.91	55.97	48.55	6.92	13.39	15.70	19.67	24.38	31.08
+ LEVER + IPS	72.73	54.17	43.64	72.73	59.03	51.71	7.27	13.58	16.40	21.52	29.24	37.10
+ SKIM (Ours)	67.05	57.95	50.68	67.05	60.96	56.41	6.74	14.50	19.33	27.20	38.04	52.39
Biased Metrics												
DEXML	75.06	41.52	28.52	75.06	78.48	80.62	55.32	79.27	86.22	91.70	95.03	97.19
+ Gandalf	74.02	41.18	28.38	74.02	77.69	79.94	54.47	78.67	85.85	91.60	95.09	97.29
+ LEVER	73.63	40.87	28.15	73.63	77.20	79.45	54.14	78.19	85.36	91.19	94.81	97.12
+ IPS	67.31	38.08	26.60	67.31	71.76	74.47	49.33	73.63	81.64	88.72	93.50	96.59
+ LEVER + IPS	69.93	39.28	27.30	69.93	74.08	76.63	51.32	75.63	83.36	89.93	94.21	96.92
+ UDAPDR	70.23	39.29	27.28	70.23	74.33	76.88	51.74	75.87	83.55	90.18	94.56	97.24
+ SKIM (Ours)	67.18	37.67	26.35	67.18	71.23	73.96	49.25	72.94	81.02	88.47	93.59	96.71
Renée	71.89	40.27	27.85	71.89	75.82	78.17	52.73	77.07	84.45	90.44	94.19	96.64
+ Gandalf	75.79	41.98	28.78	75.79	79.41	81.46	56.09	80.20	86.94	91.98	94.97	96.94
+ LEVER	77.00	42.55	29.13	77.00	80.45	82.47	56.89	81.10	87.84	92.95	95.96	97.76
+ IPS	74.00	41.17	28.24	74.00	77.71	79.76	54.62	78.60	85.32	90.40	93.47	95.61
+ LEVER + IPS	74.95	41.67	28.61	74.95	78.76	80.87	55.44	79.69	86.54	91.74	94.85	96.88
+ SKIM (Ours)	64.66	36.47	25.62	64.66	68.44	71.24	46.88	70.04	78.32	86.29	92.19	96.05

G Ablation

Adding missing labels is more effective than training with accurate propensity estimates: Propensity-based methods are the most common way of addressing missing label bias in XC. To examine their limitations, we simulate a data collection process from real world applications where a deployed model retrieves a shortlist of documents and users click on them. We study the performance of different models under varying degree of missing labels, when we have access to accurate propensities. This simulation is performed on LF-WikiTitlesHierarchy-2M dataset, where the degree of exposure of the deployed model is controlled by adjusting the top-K parameter. After top-K thresholding, a stochastic missing mechanism was applied to simulate user clicks, where a relevant query-document pair occurs (or clicked) in the dataset with probability $p_{ql} = \sigma(s(q, l))$, with $\sigma(.)$ as the sigmoid function and $s(q, l)$ as the score assigned by the pretrained `msmarco` model. Note that this simulation process is similar to click models considered in Saito et al. [88], where the probability of a click is modeled as the product of exposure and relevance probabilities (MNAR setting). Once the simulations are done, a dataset is created for every K. For every such dataset, we train Renée, Renée using IPS that make use of ground-truth p_{ql} as propensities, and finally, Renée + SKIM. Fig. 5 in the Appendix shows the effect of varying K on different models trained on that particular dataset. As seen from the graph, reweighing the Renée loss function using $p(q, l)$ (ground-truth propensities) improves the $R@100$ of the Renée model by 2%, but filling in the missing knowledge (or missing labels) using SKIM further improves $R@100$ by 13 points when $K = 200$. To further highlight the importance of having unbiased training data, we consider a training dataset based on randomly sampling an equivalent number of (query, documents) pairs from all the

Table 10: Comparison of PSP@K scores for SKIM and baseline methods on the biased test set of LF-Orcas-800K. PSP@K is reported only for the biased test set, as it serves to estimate precision on the unbiased test set. The discrepancy between unbiased precision and PSP scores on the biased test set stems from errors in propensity estimation, which are derived using the propensity model proposed in [39] with default parameter values $A = 0.55, B = 1.5$.

Method	PSP@1	PSP@3	PSP@5	PSP@10	PSP@25
DEXML	60.72	73.57	80.51	87.98	93.23
+ Gandalf	60.21	73.29	80.42	88.15	93.58
+ LEVER	59.58	72.39	79.44	87.21	92.82
+ IPS	59.01	71.17	78.09	86.01	92.16
+ LEVER + IPS	59.69	72.13	79.09	86.88	92.71
+ UDAPDR	57.78	70.30	77.51	85.89	92.49
+ SKIM (Ours)	54.30	67.11	74.80	84.12	91.62
Renée	53.77	68.79	76.90	85.69	92.01
+ Gandalf	63.41	75.29	81.40	87.69	92.15
+ LEVER	59.12	73.96	81.51	89.24	94.37
+ IPS	60.92	72.36	78.19	84.29	88.71
+ LEVER + IPS	63.47	75.23	81.31	87.66	92.12
+ SKIM (Ours)	44.91	59.93	68.87	79.73	88.92

relevant pairs known in LF-WikiTitlesHierarchy-2M (MAR setting). We see that even when 2.8% of the total data is exposed using random sampling on relevant pairs, the $R@100$ is around 37, compared to 12 when biased data is used.

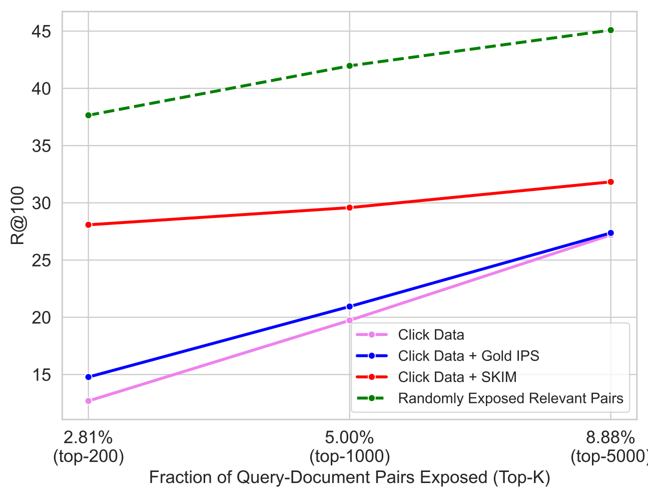


Figure 5: X axis represents increasing fraction of relevant pairs or clicks, and in brackets, we show the top-K that was used in simulation to collect that fraction of clicks. We compare Renée models trained with (a) only click data (MNAR), (b) click data + using IPS with *golden* propensities, (c) click data + SKIM and finally, (d) click data created using sampling relevant pairs uniformly at random (MAR). All four settings are compared across different fraction of relevant pairs being exposed.

Meta-data Augmented XC ablation. : We provide more details for this ablation that was discussed in the main paper (See 5.4). Just to recap, for this ablation, we wanted to test if directly using retrieval augmentation (RA) by providing available meta-data as part of the query would help in our scenario. To test this, we train a Renée model on the simulated biased training dataset of LF-WikiTitlesHierarchy-2M as used in our main experiments, however during training, we provide the metadata alongside the query. During test time too, we provide the metadata alongside the test query. We observe that this setting underperforms even when provided with the relevant knowledge for prediction. This goes to show that the Renée model memorizes the bias in the training dataset, and predicts the similar biased documents. This hints that RA

Table 11: Comparison of SKIM with baseline methods on the LF-WikiTitlesHierarchy-2M dataset. The first two sets of rows report evaluations on the unbiased test set, generated by associating Wikipedia titles with parent categories as described in [17]. The next two sets of rows present results on the simulated biased test set, where the biased ground truth is defined as the intersection of labels predicted in the top-100 by a pre-trained `ms-marco` model and the unbiased ground truth. This bias reflects the dataset creation scenario in the real-world, where only a subset of labels deemed relevant by the deployed model (i.e., judged in the top-k) is exposed to the user. For brevity, nDCG@k is abbreviated as N@k. For PSP metrics on the biased test set, refer Table 12.

Unbiased Metrics												
Method	**P@1**	**P@3**	**P@5**	**N@1**	**N@3**	**N@5**	**R@1**	**R@3**	**R@5**	**R@10**	**R@25**	**R@100**
DEXML	38.90	31.14	26.55	38.90	33.12	29.65	2.24	4.66	6.16	8.27	10.87	15.25
+ Gandalf	38.82	30.87	26.29	38.82	32.90	29.43	2.26	4.65	6.14	8.23	10.73	15.01
+ LEVER	38.41	30.24	25.47	38.41	32.23	28.56	2.05	4.31	5.69	7.57	9.89	13.78
+ IPS	36.95	29.53	25.22	36.95	31.43	28.17	2.16	4.50	5.97	8.10	10.92	15.92
+ LEVER + IPS	35.34	26.82	22.23	35.34	28.95	25.41	2.09	4.06	5.20	6.82	9.03	12.92
+ UDAPDR	33.24	25.91	22.05	33.24	27.57	24.52	1.56	3.48	4.75	6.83	10.27	17.15
+ SKIM (Ours)	43.81	36.58	32.65	43.81	38.34	35.42	2.33	5.34	7.56	11.13	16.59	25.42
Renée	39.01	31.73	26.99	39.01	33.50	29.91	1.98	4.44	5.94	7.89	9.95	13.07
+ Gandalf	40.21	32.59	27.87	40.21	34.50	30.93	2.17	4.69	6.28	8.46	11.08	15.30
+ LEVER	40.90	32.72	27.83	40.90	34.82	31.13	2.38	4.90	6.47	8.59	11.15	15.98
+ IPS	40.28	32.84	28.18	40.28	34.85	31.40	2.47	5.10	6.78	9.12	12.01	16.75
+ LEVER + IPS	40.09	32.09	27.46	40.09	34.21	30.76	2.48	5.11	6.79	9.23	12.63	19.28
+ SKIM (Ours)	41.53	35.71	32.42	41.53	37.14	34.71	2.19	5.10	7.31	11.18	17.55	28.32
Biased Metrics												
DEXML	35.98	27.49	22.40	35.98	34.53	34.48	14.20	25.47	31.01	36.82	40.91	44.22
+ Gandalf	35.63	26.87	21.82	35.63	33.86	33.70	14.02	24.83	30.12	35.74	39.84	43.32
+ LEVER	35.13	26.20	21.07	35.13	33.17	32.89	13.81	24.33	29.41	34.94	39.31	42.99
+ IPS	34.20	25.98	21.19	34.20	32.74	32.72	13.48	24.12	29.49	35.50	40.19	43.98
+ LEVER + IPS	32.68	23.82	19.01	32.68	30.40	30.01	12.74	22.13	26.75	32.24	37.50	42.35
+ UDAPDR	28.28	19.90	15.70	28.28	25.67	25.08	10.94	18.32	21.95	26.54	31.47	37.16
+ SKIM (Ours)	25.76	18.95	15.38	25.76	23.46	23.28	8.93	16.60	20.91	26.44	32.21	37.64
Renée	35.15	26.88	21.73	35.15	33.56	33.30	13.57	24.53	29.69	34.93	38.56	41.62
+ Gandalf	35.10	26.50	21.35	35.10	33.30	33.02	13.65	24.40	29.47	34.84	38.91	42.20
+ LEVER	35.71	26.94	21.76	35.71	33.90	33.68	13.93	24.88	30.14	35.78	40.05	43.86
+ IPS	34.49	26.46	21.45	34.49	33.04	32.89	13.34	24.26	29.50	34.90	38.59	41.19
+ LEVER + IPS	34.78	26.01	21.00	34.78	32.94	32.75	13.71	24.26	29.40	35.01	39.48	43.54
+ SKIM (Ours)	19.41	15.24	12.88	19.41	17.94	18.12	5.81	12.30	16.47	22.19	28.70	35.05

like works might fail if used with biased training datasets. However, if we train the Renée model, again with metadata alongside the query, but on training dataset generated by SKIM, we observe a sharp increase in performance. In this setting, the Renée + SKIM model actually starts generalizing with the metadata present alongside the test query, and performs better as compared to the Renée + SKIM model that uses only the test query during test time (see table 4, R@100, in case of RA used with SKIM, increases from 28.3 to 45.98 i.e. by more than 15 absolute points). This shows that RA works might be complementary to our proposed approach.

Effect of number of synthetic pairs (new training pairs) on unbiased performance. : In figure 7, we observe that as we increase the number of new training pairs or the synthetic pairs obtained using SKIM, we see an increase in downstream unbiased performance of Renée model, as expected. We control the number of added pairs using the similarity threshold τ. However, when we go pass a certain limit of number of added pairs, we see that the performance drops. This might be due to the fact that noisy pairs are being added in the dataset due to the low value of τ. Thus, there is a sweet-spot when we obtain the maximum unbiased performance using SKIM. This drop in performance could be mitigated by using a better retriever for mapping, or by increasing the number of synthetic queries that is being generated by the SLM during Step 1 of SKIM.

Table 12: Comparison of PSP@K scores for SKIM and baseline methods on the biased test set of LF-WikiTitlesHierarchy-2M. PSP@K is reported only for the biased test set, as it serves to estimate precision on the unbiased test set. The discrepancy between unbiased precision and PSP scores on the biased test set stems from errors in propensity estimation, which are derived using the propensity model proposed in [39] with default parameter values $A = 0.55, B = 1.5$.

Method	PSP@1	PSP@3	PSP@5	PSP@10	PSP@25
DEXML	50.51	56.97	63.44	74.14	85.34
+ Gandalf	51.26	56.07	61.89	72.00	82.97
+ LEVER	44.81	50.11	55.99	66.71	79.53
+ IPS	57.11	61.41	65.93	74.24	84.78
+ LEVER + IPS	52.62	55.59	59.25	66.68	78.69
+ UDAPDR	47.99	49.94	52.73	58.98	70.35
+ SKIM (Ours)	33.97	37.47	42.05	51.35	65.79
Renée	33.81	42.78	50.05	61.48	72.16
+ Gandalf	36.61	44.30	50.65	61.22	72.57
+ LEVER	40.40	48.29	55.66	68.23	81.04
+ IPS	39.83	46.77	51.94	60.17	68.74
+ LEVER + IPS	49.67	54.38	59.45	68.77	79.98
+ SKIM (Ours)	15.67	20.92	25.67	35.03	51.55

Justification of design choices in SLM. : We perform an ablation to justify our design choices regarding (i) finetuning the SLM, and (ii) providing metadata to the SLM. In Figure 6, we compare a pretrained SLM, finetuned SLM and the LLM performance on the task of good *quality* synthetic document generation (measured using GT coverage @ 100, see figure 6). SLM used is Llama2-7B and LLM is GPT4. We see a clear increase in performance when we move from pretrained SLM, to finetuned SLM and then finally to LLM. Additionally, we observe that metadata consistently improves generation quality when used with any model. Interestingly, finetuned SLM reaches close to LLM performance when provided with metadata. Thus, SKIM is able to generate LLM-like quality of synthetic documents, with a much higher throughput, when finetuned and provided with relevant metadata, even though it is unstructured.

Size of the SLM used in SKIM. : In table 5, we see the effect of the size of the SLM used in Step 1 of SKIM algorithm. We observe that even if we decrease the size of the SLM almost by 7x, we see marginal drop (< 1%) in downstream performance. This highlights extreme scalability of SKIM even when used in limited compute regimes. How far can this be stretched is an interesting future work.

H Related Works (Extended Version)

Extreme classification: Extreme Classification or XC is a prominent supervised learning formulation for large-scale (in orders of millions of documents) retrieval problems and has been very influential due to its success in many practical scenarios. The methods proposed for extreme classification can be broadly categorized into two families: one-versus-all classifier-based methods [39, 76] and dual encoders [23, 33, 34, 40, 49]. Key innovations in this domain include transition from sparse feature based classifier learning [6, 9–11, 15, 38, 39, 48, 74–76] to deep-encoder based representations [23–25], efficient end-to-end training frameworks [40], Siamese networks for query/document representations [58, 59, 78], exploration of more advanced deep learning architectures for the encoder and the classifier components [22, 24, 33, 49, 102, 104, 105], and effective negative sampling strategies for training [23, 32, 44, 82, 83, 101].

Missing label and long-tail biases: XC models, as other retrieval models, are susceptible to biases in their training set collected from system log data. Key biases that significantly affect the training and evaluation of XC models include document distribution bias [17, 91], selection bias[63], position bias[19], exposure bias [53, 106] and inductive bias (refer to [18] for a recent survey). Collectively, these biases lead to the *missing label* problem wherein some relevant documents for a query are missing not at random. Additionally, certain documents are less represented than others in the training data, leading to a long-tail distribution over documents. Consequently, missing labels and long-tail biases presents two critical challenges in XC [91]: 1) relevant documents goes missing in the observed training data; 2) infrequent (tail) documents are much harder to predict than frequent (head) documents due to data imbalance. In Section 3, we formalize how the systematic nature of missingness due to production systems leads not only to missing label bias but also to missing knowledge in training data.

Addressing missing label and long-tail biases: Within the XC literature, the most commonly adopted solution to the missing labels problem is propensity-based learning [39, 77, 97, 99]. The propensity score estimates the likelihood for a query-document pair to go missing, given that the document is relevant. The standard loss functions and metrics can be corrected for the missing label bias by reweighting the individual terms by the propensity values. Assuming that the propensities only depend on the document and that these values are known or

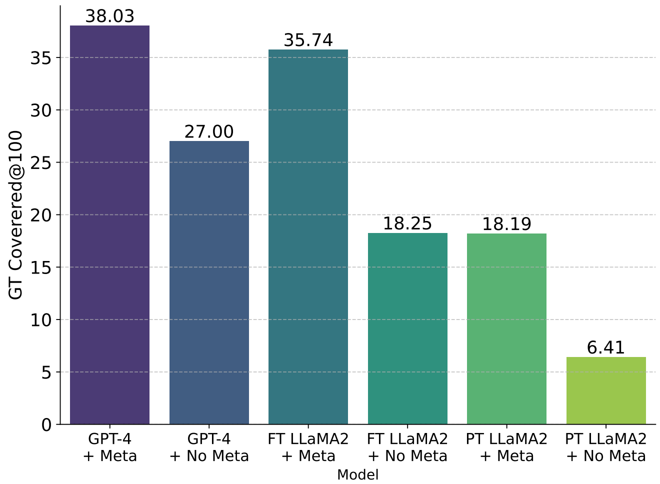


Figure 6: Comparison of the effect of different language models and metadata on the quality of generated synthetic documents. GPT-4, a pretrained LLaMA2-7B, and a finetuned LLaMA2-7B are evaluated with and without metadata on the LF-WikiTitlesHierarchy-2M dataset where synthetic document generation is performed. The quality of synthetic documents is measured by GT covered @ 100, which is the average number of actual ground-truth (GT) documents retrieved using synthetic documents. For each synthetic document, the top 100 nearest documents are retrieved using a finetuned NGAME encoder and checked against the GT. The pretrained LLaMA2 has the lowest quality, while the finetuned LLaMA2 with metadata almost matches GPT-4 with metadata (LLaMA2: 35.7 vs. GPT-4: 38.0).

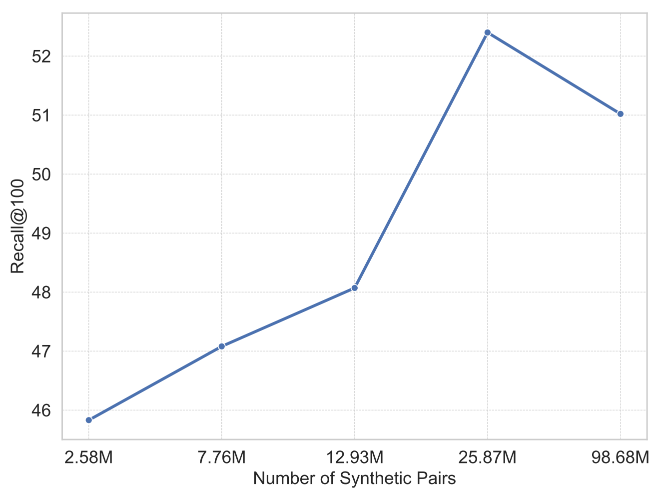


Figure 7: Effect of the number of synthetic pairs added in Step 2 of SKIM on the downstream retrieval performance of Renée on the LF-Orcas-800K dataset. We control the number of synthetic pairs added by tuning the similarity threshold τ.

can be estimated from some external meta-data [64], Jain et al. [39] introduced unbiased loss functions and various evaluation metrics like nDCG@K, Recall@K and Precision@K. But, as we show in this paper, both theoretically and empirically, the missing label bias in XC can cause *missing knowledge* problem which cannot be recovered by propensity based learning. Evaluation of XC models on unbiased datasets using propensity scored metrics, such as PSP@k, PSN@k [14, 39], will also suffer from the same limitations.

In similar settings such as recommendation systems [88, 89, 95], and positive unlabeled learning (PUL) [12, 13, 41, 47], the missing labels problem has been addressed by propensity based methods [39, 77], imputation error based methods [17, 50], and doubly robust methods [51, 55, 95] that try to combine both these paradigms. Studies addressing this problem for recommendation systems using propensity-based corrections mechanisms [45, 88, 90] often require an unbiased subset of training data over the user-item space to train the propensity models

Table 13: Comparative analysis of the click-based ground truth (row 1), a state-of-the-art XC algorithm, Renée + IPS [77] (row 2) Renée + LEVER (row 3), and our method, Renée + SKIM (row 4). Entries in black represent documents that have a click in the biased ground truth (row 1) or documents that are predicted by the models in top-100 predictions (rows 2,3 and 3). Grey cells indicate documents that are relevant to the query "Exon Definition" but are absent from both the ground truth (in row 1) or absent the top-100 predictions of the models (in rows 2, 3, and 4). All documents are relevant to the query "Exon Definition," according to the unbiased test set. Documents are grouped into four clusters based on their related concepts. The biased clicks in row 1 connect the query to document concepts such as "Exon" and "Exome Sequencing." However, there are no clicks linking "Exon Definition" to concepts like "Genes" or "RNA." The IPS and LEVER algorithm (rows 2,3) retrieves documents from clusters similar to those seen during training (e.g., "Exon" and "Exome Sequencing") but struggles with other concepts. In contrast, SKIM (row 4) successfully retrieves documents across all four clusters, emphasizing the value of integrating world knowledge. Document ranks are shown next to the URL in bold, and URLs have been shortened for brevity.

		Document Concepts			
	Query	Exon	Exome Sequencing	Genes	RNA
Biased Ground Truth	exon definition	①merriam-webster.com dictionary/exon, ②en.wikipedia.org wiki/Exon ③net.science/diff-bw -exons-and-introns	①en.wikipedia.org/ Exome-sequencing, ②broadinstitute.org /what-exome-seq, ③blogs.scientific american.com/10-things-exome-seq..	①news-medical.net /What-are-Genes ②www.genomenewsnetwork /whats a-genome/ ③en.wikipedia.org/wiki /Human-genome ④en.wikipedia.org/wiki /Gene-expression	①en.wikipedia.org/wiki /Messenger-RNA ②en.wikipedia.org/wiki /Precursor-mRNA ③en.wikipedia.org/wiki /RNA-splic.. ④en.wikipedia.org/wiki/ Alternative-splice
Renée + IPS [77] Predictions (top-100)	exon definition	①merriam-webster.com dictionary/exon **(2)**, ②en.wikipedia.org wiki/Exon **(1)** ③net.science/diff-bw -exons-and-introns	①en.wikipedia.org/ Exome-sequencing ②broadinstitute.org /what-exome-seq ③blogs.scientific american.com/10-things-exome-seq..	①news-medical.net /What-are-Genes ②www.genomenewsnetwork /whats a-genome/ ③en.wikipedia.org/wiki /Human-genome ④en.wikipedia.org/wiki /Gene-expression	①en.wikipedia.org/wiki /Messenger-RNA ②en.wikipedia.org/wiki /Precursor-mRNA ③en.wikipedia.org/wiki /RNA-splic.. ④en.wikipedia.org/wiki/ Alternative-splic..
Renée + LEVER Predictions (top-100)	exon definition	①merriam-webster.com dictionary/exon **(2)**, ②en.wikipedia.org wiki/Exon **(1)** ③net.science/diff-bw -exons-and-introns **(15)**	①en.wikipedia.org/ Exome-sequencing **(12)**, ②broadinstitute.org /what-exome-seq **(51)**, ③blogs.scientific american.com/10-things-exome-seq.. **(91)**	①news-medical.net /What-are-Genes ②www.genomenewsnetwork /whats a-genome/ ③en.wikipedia.org/wiki /Human-genome ④en.wikipedia.org/wiki /Gene-expression	①en.wikipedia.org/wiki /Messenger-RNA ②en.wikipedia.org/wiki /Precursor-mRNA ③en.wikipedia.org/wiki /RNA-splic.. ④en.wikipedia.org/wiki/ Alternative-splic..
Renée + SKIM predictions (top-100)	exon definition	①merriam-webster.com dictionary/exon **(4)**, ②en.wikipedia.org wiki/Exon **(1)** ③net.science/diff-bw -exons-and-introns **(10)**	①en.wikipedia.org/ Exome-sequencing **(18)**, ②broadinstitute.org /what-exome-seq **(11)**, ③blogs.scientific american.com/10-things-exome-seq.. **(35)**	①news-medical.net /What-are-Genes **(82)** ②www.genomenewsnetwork /whats a-genome/ **(20)** ③en.wikipedia.org/wiki /Human-genome **(80)** ④en.wikipedia.org/wiki /Gene-expression **(52)**	①en.wikipedia.org/wiki /Messenger-RNA **(61)** ②en.wikipedia.org/wiki /Precursor-mRNA **(73)** ③en.wikipedia.org/wiki /RNA-splic.. **(12)** ④en.wikipedia.org/wiki/ Alternative-splic.. **(16)**

[90, 96]. However, obtaining such unbiased subsets for large-scale retrieval applications is difficult and their absence may lead to high variance in estimated propensities.

Teacher Models and Data Augmentation: There is now a significant body of work that tries to train performant XC or retrieval models by augmenting [16, 27, 43, 86] the training dataset with external resources like teacher models, and query/document meta-data. Although these approaches don't make explicit connections to the problem of missing label bias, as show in Section 3, they are relevant to solving the missing labels problem. Recent efforts [67, 69] have shown that auxiliary information or structured metadata linked with the queries or the documents, though rarely available with XC or short-text retrieval datasets, can be utilized to boost performance of XC models. LEVER [17] and Gandalf [50] uses a teacher-model trained on the existing dataset to help XC models enhance their performance on tail documents. RocketQA [103] uses a more expressive cross-encoder model to impute the training data for a student dual-encoder retrieval model. SemsupXC [5] is another closely related work that attempts to incorporate external information by scraping information from the web in order to improve performance in zero-shot or few-shot settings, involving previously unseen documents. But, as their model is trained on the biased training data, the trained model will continue to suffer from missing knowledge issue even with the help of additional meta-data.

Recently, LLMs have been used for generating data for training task-specific models [52], SLMs/LLMs [31, 36, 61, 84, 85], and multimodal models [56]. This approach to training models using LLM-generated datasets may help in combating various problems like data scarcity in low-resource settings, lack of unbiased datasets, noisy and fake information in web-based training data, or where annotating sufficient data is a costly process [4, 57]. The strategies used for this kind of dataset creation approaches include heuristic filtering [79], quality filtering [2, 29, 73], deduplication [1], data mixing [7, 28, 100], synthetic data generation [31, 61], data augmentation [98], or generating task-specific synthetic data by transforming existing datasets for a similar task [30]. There have also been some efforts to directly finetune SLMs as encoders for query / documents in a retrieval task [60, 70, 72].

In contrast, we propose a way for *completing* existing task specific dataset by addressing *knowledge gaps* present in the dataset. While existing work mainly focus on creating task-specific datasets from scratch or augmenting to extend the scale of these datasets, it is possible that the biases in the source dataset / LLM may just get scaled up in this synthetic data creation process. In addition, using oracles LLMs like GPT-4 or Claude is not scalable, so there is a need for more efficient ways to augment the training data.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009