Extreme Meta-Classification for Large-Scale Zero-Shot Retrieval

Anonymous Author(s)*

ABSTRACT

We aim to develop accurate and efficient solutions for large-scale retrieval tasks where novel (*zero-shot*) items can arrive continuously at a rapid pace. Conventional Siamese-style approaches embed both queries and items through a small encoder and retrieve the items lying closest to the query. While this approach allows efficient addition and retrieval of novel items, the small encoder lacks sufficient capacity for the necessary world knowledge in complex retrieval tasks. The extreme classification approaches have addressed this by learning a separate classifier for each item observed in the training set which significantly increases the representation capacity of the model. Such classifiers outperform Siamese approaches on observed items, but cannot be trained for novel items due to data and latency constraints. To bridge these gaps, this paper develops: (1) A **new algorithmic framework**, EMMETT, which efficiently synthesizes classifiers for novel items, on-the-fly, by relying on the readily available classifiers for observed items, (2) A **new algorithm**, IRENE, which is a simple and effective instantiation of EMMETT that is specifically suited for large-scale deployments, and (3) A **new theoretical framework** for analyzing the generalization performance in large-scale zero-shot retrieval which guides our algorithm and training related design decisions.

Comprehensive experiments are conducted on a wide range of retrieval tasks which demonstrate that IRENE improves the zero-shot retrieval accuracy by up to 15% points in Recall@10 when added on top of leading encoders. Additionally, on an online A/B test in a large-scale ad retrieval task in a major search engine, IRENE improved the ad click-through rate by 4.2%. Lastly, we validate our design choices through extensive ablative experiments. The code will be released publicly upon paper acceptance.

CCS CONCEPTS

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

KEYWORDS

extreme classification, large-scale retrieval, sponsored search advertising

1 INTRODUCTION

Large-scale retrieval involves retrieving the items relevant to a query from a pool of hundreds of millions of candidate items. Such tasks frequently arise in modern-day web applications such as web search [7], computational advertising [3], product recommendation [10], and so on. To optimize user satisfaction, the retrieved results need to be highly relevant to the query and delivered in real-time, typically within milliseconds. Additionally, due to the exponential growth of digital content, novel items get introduced into these systems in vast quantities daily, which need to be swiftly processed and inserted into the candidate pool to ensure up-to-date results. This paper aims to develop highly accurate and efficient solutions for problems of large-scale text-based retrieval with *novel* items (also referred to as *zero-shot* items). Note that this scenario is different from the one considered in [16, 45] where the task itself is zero-shot.

A widely used technique for large-scale retrieval is dense retrieval [50] where both queries and items are represented as embeddings in a shared low-dimensional space such that the items relevant to a query are positioned closer to it than the irrelevant ones. The relevant items, typically few in number, are then retrieved in almost real-time using scalable Approximate Nearest Neighbour Search (ANNS) indices [21]. The accuracy of retrieved results depends on the quality of the derived query and item representations.

Typical dense retrievers are based on a Siamese encoder architecture which uses a common deep neural encoder to derive both query and item representations from their raw text inputs [24, 46]. Usually, a small and efficient encoder is utilized to reduce the representation latencies, which enables large-scale deployments and quick insertion of novel items. However, a small encoder often lacks the capacity to model the complexity inherent in retrieval tasks [15]. For instance, item descriptions frequently contain named entities, numbers, model numbers, and ambiguous phrases with issues of synonymy and polysemy whose resolution requires extensive world knowledge that cannot be contained within a small encoder. As a result, the representations from these approaches are inferior and degrade the retrieval performance.

Recently, Extreme Classification (XC) methods have emerged as promising alternatives for improved dense retrieval. Leading extreme classifiers augment a small Siamese encoder with a massive linear classifier layer at its output which significantly boosts the model capacity [12, 13]. Each item *observed* in the training set is endowed with its own classifier, which absorbs the world knowledge pertinent to the item when trained from the historical click logs. During retrieval, a query is first passed through the encoder and then projected onto its relevant items by applying classifiers. When there are enough clicked query samples for training, the classifier-based item representations can be more precise than the text-restricted representations from a small encoder. However, zero-shot retrieval is not supported, as classifiers cannot be trained

for novel items with no clicked samples. Even if we could, learning new classifiers from scratch is time-consuming and delays the representation of novel items, making the candidate pool outdated.

This paper addresses the limitations of the existing approaches on large-scale zero-shot retrieval. Our primary research question is: *How to construct accurate representations for novel items without significant computational overhead in large-scale retrieval tasks?*

We propose a novel algorithmic framework, Extreme Meta Classification (EMMETT), as an answer to this question. EMMETT extends conventional Extreme Classification to zero-short scenarios. Often in large-scale retrieval tasks, millions of observed items with associated user-clicked queries are available for training from historical logs. This leads to two key insights that underpin EMMETT's design:

First, ignoring the shifts in item distribution, extreme classifiers trained for a million observed items are likely to contain most of the world knowledge required for any novel item in a distilled and readily usable form. With this intuition, EMMETT builds a pipeline with two main modules: (1) a Classifier Selector (CS) module, which takes a novel item as input and rapidly shortlists a few observed item classifiers that are most informative for it, by using efficient filtering techniques, and (2) a Meta-classifier Generator (MCG) module, which combines these shortlisted classifiers to synthesize a novel item's classifier with minimal latency. We call such a classifier derived from other classifiers as a *meta-classifier*.

Second, a million observed items can also serve as a large number of samples for optimally training the EMMETT modules. With a careful loss design, each observed item can be treated as a proxy for a zero-shot item which yields a massive training set for zero-shot retrieval. Training EMMETT on such a dataset can ensure robust generalization on novel items. We theoretically and empirically validate this claim in Sections **5** and **7**, respectively.

Figure **1** presents the architectural overview of EMMETT. EMMETT is a generic framework where different choices for the two modules and the training strategy can potentially yield algorithms with varying accuracy and efficiency trade-offs, which also presents ample opportunities for future research explorations.

This paper develops one such algorithmic instantiation of the EMMETT framework, named IRENE. Given a generic Siamese encoder augmented with extreme classifiers on observed items, IRENE is designed to add meta-classifier functionality to this base with minimal effort. IRENE selects the observed item classifiers by using efficient ANNS-search, where the item representations are derived from the base Siamese encoder itself. This avoids training an additional module thus reducing the training and deployment efforts, but can also add noisy classifiers to the shortlist. To effectively filter the noise, combine the useful knowledge and synthesize meta-classifiers, IRENE trains a separate Meta-Classifier Generator model that is based on a powerful transformer architecture. By using a small classifier shortlist, the meta-classifier synthesis can be highly efficient as it requires only one transformer pass. IRENE training freezes the Siamese encoder and extreme classifiers, and only trains the Generator using a weighted One-vs-All classification loss. Both novel item representation and retrieval are real-time and require few milliseconds in IRENE. Figure **2** depicts the overall architecture of IRENE.

Theoretical analysis is a crucial tool which can guide the design of robust and effective machine learning systems. For this purpose, we develop a novel theoretical framework for analyzing the generalization performance in large-scale zero-shot retrieval. It is based on a key insight that, given a dense retriever that outputs calibrated query-item relevance scores, the zero-shot retrieval task is equivalent to a binary classification task with a query-item pair as our data sample. This equivalence allows us to leverage and further extend the theoretical ideas from traditional binary classification literature. Our results in Section **5** show that (1) One-vs-All loss in IRENE leads to strong zero-shot generalization performance by reducing the variance in the training loss, (2) Freezing extreme classifiers during IRENE avoids overfitting and is therefore crucial for robust zero-shot generalization, (3) Number of observed items and the Generator's complexity also affect the zero-shot generalization. These results are also validated empirically in Section **7**.

This paper makes the following key contributions:

(1) A **novel and generic algorithmic framework**, EMMETT, for learning accurate representations of novel items, thus improving the zero-shot retrieval performance.
(2) A **novel and efficient algorithm**, IRENE, that can be added to any Siamese encoder and is well-suited for large-scale real-world applications
(3) A **novel theoretical framework** for analyzing the generalization performance in large-scale zero-shot retrieval
(4) **Comprehensive experimental validation** demonstrated that IRENE can offer up to 15% gains with minimal overheads, in zero-shot retrieval on diverse benchmark datasets varying in their scale, application and dataset properties.
(5) **Online A/B tests** on a large-scale Sponsored Search application in a major search engine showed that IRENE can improve ad click rate by 4% thus demonstrating its real-world utility.

2 RELATED WORK

This section provides a comprehensive review of the past literature that is pertinant to our work. It presents the different classes of approaches available for retrieval and highlights their scalability, accuracy and zero-shot capabilities.

Traditional statistical approaches: Traditional approaches to large-scale retrieval, including TF-IDF [23] and BM25 [35], relied on measuring the co-occurrences of terms in a query text and an item text. These methods utilized inverted term indices for efficient item search which scaled well and allowed rapid insertion of novel items. However, they lacked the ability to understand the context of the inputs. ZestXML [17], a recent method for large-scale zero-shot retrieval, extended the term-matching to also account for semantic correlations between the terms which improved the retrieval performance. Nevertheless, these traditional methods have been largely superseded by semantic matching techniques using deep networks, which offer superior accuracy.

Siamese-encoder approaches: These approaches employ deep neural encoders to embed both query and item texts into a shared, low-dimensional representation space. Leading approaches such as DPR [24], ANCE [46], MACLR [47], RocketQA [34] and NGAME [13], utilize a transformer-based encoder. ANCE, RocketQA and NGAME

also implement advanced negative-mining strategies for robust encoder training. MACLR leverages self-supervised close-task based learning for pre-training. In these approaches, incorporating a novel item is efficient and requires just a single encoder pass over their raw features. However, to maintain real-time response rates, these methods are constrained to use a small transformer, which degrades retrieval accuracy due to lack of world knowledge necessary for large-scale retrieval. Recent works like ColBERT [27] and Semsup-XC [2] modify the Siamese architecture to permit token-level interactions between the query and item texts which improves retrieval accuracy. However, this enhancement also increases inference latency, making these models less suitable for practical deployments. Semsup-XC also relies on web-scraped meta-data to boost performance on unseen items which can be expensive.

Extreme classification approaches: These approaches learn one-vs-all classifiers to represent each observed item in the training set [14, 15, 18, 19, 26, 31, 33, 38, 49]. These are highly accurate, offer low inference latencies and have been successfully applied to a wide range of large-scale retrieval tasks including document tagging [6], product-to-product recommendation [22, 25], and sponsored search [12, 13]. Unfortunately, most of these approaches do not have support for zero-shot items.

Among these, the most relevant to our work is DEXA [15]. DEXA learns a small set of classifiers, one classifier per cluster of items to improve the retrieval performance. DEXA can be viewed as a naive version of EMMETT with an inferior architecture and training loss, and is significantly outperformed by IRENE (see Section **6**).

Other zero-shot retrieval approaches: Several approaches have been proposed for zero-shot multi-label retrieval such as ESZSL [36], COSTA [32], LAF [29] *etc.* These works consider a few thousand labels (items) and typically do not scale to million items.

Over the past year, large language model based zero-shot retrieval is also gaining popularity [39]. These models are expensive and are not yet demonstrated for large-scale retrieval.

The works in [16, 45] study a different zero-shot setting where the task itself is novel with no available relevance signals for model training. They propose techniques based on domain transfer as solutions. Unlike these works, our setting assumes that abundant click data is available as relevance signals and focusses on leveraging them to improve the retrieval with zero-shot items.

3 EXTREME META-CLASSIFICATION

In this section, we introduce **EMMETT**, our proposed ExtreMe METa-classificaTion framework for improving zero-shot retrieval by means of incorporating the knowledge present in the learnt classifiers of observed items.

3.1 Preliminaries

We describe the notation and background that is necessary for the rest of this section. Consider queries X drawn from the space $\mathcal{X}$, which are mapped to items Z drawn from the item space $\mathcal{Z}$. During training, we have access to N data points $\{X_i\}_{i=1}^{N}$ and L *observed* items $\{Z_\ell\}_{\ell=1}^{L}$. Each data-item pair is associated with a label $y_{i\ell} = \{0, 1\}$, which is 1 when the item is relevant to the input query, and 0 when the item is irrelevant to the query.
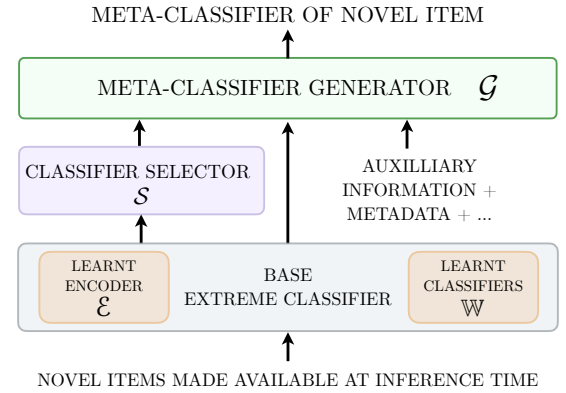

Figure 1: An overview of our proposed ExtreMe METa-classificaTion (EMMETT) framework. Given an extreme classification base (encoder $\mathcal{E}$ and classifiers $\mathbb{W}$), EMMETT consists of two modules, where (a) the classifier selector $\mathcal{S}$ retrieves the *most informative* classifiers for a novel item, and (b) the meta-classifier generator $\mathcal{G}$ combines the selected classifiers and other meta-data to create the meta-classifier.

Dense retrieval (DR) algorithms consider a deep feature encoder $\mathcal{E}$ that project the data samples and items to the space of d-dimensional encoder representations, denoted as $x = \mathcal{E}(X)$ and $z = \mathcal{E}(Z)$, respectively, both $\mathcal{E}(\mathcal{Z}), \mathcal{E}(\mathcal{X})$ belonging to $\mathbb{R}^d$. The query and observed-item sets in this encoder space are represented as $\mathbb{X} = \{x_i\}_{i=1}^{N}$, and $\mathbb{Z} = \{z_\ell\}_{\ell=1}^{L}$, respectively. In Siamese DR algorithms, both query and item representations share the same encoder space. Items are retrieved for a given query, typically using approximate nearest neighbor search (ANNS), based on the computed similarity score $x_i^\top z_\ell$.

Extreme classification (XC) algorithms are built on the intuition that classifier-based item representations can surpass the performance of encoder-representations. In XC, each item z_ℓ is represented by a 1-vs-all classifier w_ℓ, collectively denoted as $\mathbb{W} = \{w_\ell\}_{\ell=1}^{L}$. These are trained using triplet-based [13] or cross-entropy-based [20] losses atop the encoder representation. The similarity score is calculated between the query representation and the classifiers , given by $x_i^\top w_\ell$. Thus, the XC module comprises an encoder and a set of learned classifiers $(\mathcal{E}, \mathbb{W})$. While effective in retrieving observed items, these methods face challenges in retrieving novel items.

3.2 The EMMETT Framework

We propose EMMETT, an **extreme meta-classification** framework for zero-shot generalization in the retrieval setting. The EMMETT framework is designed to develop highly accurate and efficient retrieval models capable of handling novel item retrieval during inference. Existing XC models, despite their accuracy through classifier-based approaches built atop encoder representations, suffer from high data and resource requirement and therefore, and do not generalize to the zero-shot setting.

The EMMETT framework considers training data $\mathbb{X}$ and $\mathbb{Z}$, a **base extreme classifier** $(\mathcal{E}, \mathbb{W})$, comprising the encoder $\mathcal{E}$ and

observed-item classifiers $\mathbb{W}$, and a set of novel items $\mathbb{Z}_n = \{z_\ell\}_{\ell=1}^{L_n}$ made available at inference time, EMMETT includes two modules:

- **Classifier selector $\mathcal{S}$**: This module selects the *most informative* set of classifiers from the base XC model for a novel item $z \in \mathbb{Z}_n$ introduced at inference time.
- **Meta-classifier generator $\mathcal{G}$**: This module combines the classifiers selected by $\mathcal{S}$, and other auxiliary information (such as the encoder representation z) to create the meta classifier u associated with item z.

EMMETT offers flexibility in its implementation through a variety of design choices, such as the, base XC framework, choice of encoder architecture and training methods. The classifier selector's $\mathcal{S}$ design involves choosing a selection algorithm and determining the number of classifiers to select. An ideal selection algorithm should choose *informative* classifiers, taking into account factors such as named entities, numbers, ambiguous phrases, synonyms, etc. Since the task of classifier selection can be viewed as retrieval, any existing retrieval methods can be used. In the context of EMMETT we prioritize efficient classifier selection and quick incorporation of novel items. The generator's $\mathcal{G}$ design must take into consideration model complexity (which we discuss in Section 5), ensuring it effectively incorporates the selected classifiers and potential auxiliary data to learn the meta-classifier u without overfitting. Another component of EMMETT is in designing loss functions for towards zero-shot generalization, and choosing appropriate training strategies, such as end-to-end, modular, etc.

Figure 1 provides a visual illustration of these three components. EMMETT enables interpreting existing XC models' zero-shot generalizability. For example, in DEXA [15], given a novel item, its classifier can be selected based on cluster assignment, and then summed with its encoder representation. Similarly, in NGAME [13], the classifier selector is an indicator function, which returns a classifier only for the observed items. NGAME's decision tree, built using the encoder representation and classifier, can be viewed as its generator block. DEXA's and NGAMe's relatively poorer zero-shot generalization can be attributed to their simplistic $\mathcal{S}$ and $\mathcal{G}$ architectures. An additional reason why these models perform poorly on novel items can traced back to their loss functions, which are not tailored for zero-shot performance.

In Section 4, we present IRENE, a specific instance of EMMETT designed for zero-shot generalization. This includes a thoughtful design of classifier selector and generator modules, considering training strategies, deployment ease, new item incorporation at inference, and loss functions prioritizing zero-shot performance. Theoretical analysis of IRENE's effectiveness is detailed in Section 5.

4 THE IRENE EXTREME META-CLASSIFIER

We now present the **IRENE** algorithm, our proposed approach for Improved REtrieval of Novel itEms with extreme meta-classification. IRENE is adaptable to any XC framework. Specifically, we utilize a 6-layer DistilBERT encoder, trained with state-of-the-art, computationally efficient algorithms such as NGAME [13], ANCE [46], MACLR [47], and DPR [24]. Given a real-world application such as product-to-product recommendations, document tagging, or matching user queries to advertiser keywords appropriate feature encoders may be chosen. After training, the encoder $\mathcal{E}$ and the

observed-item classifiers $\mathbb{W}$ remain fixed through subsequent stages. The learnt classifiers are accessible via a lookup function C_{lf} based on the items' encoder representations, *i.e.*, $w_\ell = C_{lf}(z_\ell)$.

The classifier selector $\mathcal{S}$, for any item z_ℓ, either observed or novel, retrieves K classifiers of related items using an ANNS index built on the item representations $\mathcal{E}(\mathbb{Z}_s)$, served via approaches such as DiskANN [21]. The selection is based on a maximum inner product search (MIPS) between the novel item, and observed items $\arg\max_K\{z_\ell^\top z_o; \forall z_o \in \mathbb{Z}\}$. The complexity of these approaches has typically been shown to be logarithmic in the number of items, *i.e.*, $O(() \log L)$. The choice of K, a key hyper-parameter, balances model capacity and complexity, which we formalize in Section 5 and validate in Section 7. Empirically, we found $K \approx 3$ to work well. **The meta-classifier Generator $\mathcal{G}_{\phi}$**: IRENE's generator is based on a transformer architecture, inspired by the capability to be universal approximators [48]. While the transformer layers are capable of handling a variety of input embeddings, in IRENE, the input sequence consists of the selected classifiers and the novel item's encoder representation and is passed through self-attention and linear blocks. In particular, the meta classifier is given by $u_\ell = \mathcal{G}_{\phi}(z_\ell, \mathcal{S}(z_\ell))$ and each layer can be expressed as as:

$$\text{Linear(Self-Attention}(z_\ell + t_{enc}, w_\ell^1 + t_{clf}, w_\ell^2 + t_{clf}, \ldots), \quad (1)$$

where $w_\ell \in \mathcal{S}(z_\ell)$, t_{enc} and t_{clf} are learnable *type embeddings* that distinguish between encoder and classifier inputs. The transformer layer is capable of learning correlations between the classifiers and the encoder representation, yields superior meta-classifiers over methods that consider naive weighted summation for the generator (cf. Section 7). The computation complexity is $O((K+1)^2 d)$ for the self attention layer [42], and $O(K+1)$ for the linear layer. Figure 2 illustrates the modules present in the IRENE framework.

4.1 IRENE: Training and Inference

In this section, we outline the training strategy for IRENE. Firstly, we **assume that the representation encoder $\mathcal{E}$, and the set of observed-item classifiers $\mathbb{W} = \{w_\ell\}_{\ell=1}^L$ are made available to us apriori, and remain fixed.** With these components $(\mathcal{E}, \mathbb{W})$, the ANNS index is constructed over the encoder representations of the observed items $\mathbb{Z}$. Classifier selection follows the methodology detailed in Section 4.

IRENE's meta-classifier training employs a binary cross-entropy loss reformulated for zero-shot performance. In particular, to simulate novel items at training time, given an observed item z_ℓ, instead of using the item's own classifier w_ℓ, the meta-classifier u_ℓ, derived from the selected classifiers, is utilized. That is, an observed item its trained with its selected *neighboring* classifiers and not its own. The original targets $y_{i\ell}$ remain unchanged. The loss is given by:

$$\mathcal{L}_{\mathcal{G}}(\phi) = \sum_{x_i \in \mathbb{X}} \sum_{z_\ell \in \mathbb{Z}} \left(Cy_{i\ell} \ln\left(x_i^\top u_\ell\right) + (1-y_{i\ell}) \ln\left(1 - x_i^\top u_\ell\right)\right), \quad (2)$$

where $u_\ell = \mathcal{G}_{\phi}(z_\ell, \mathcal{S}(z_\ell))$, $x = \mathcal{E}(X)$, and C is the weight for misclassified positive query-item pairs. This weight addresses the imbalance between positive and negative query-item pairs in standard XC settings, as misclassifying positives significantly impacts performance. [4]. We link the likelihood of encountering positive-pairs to zero-shot generalization performance in Section 5.
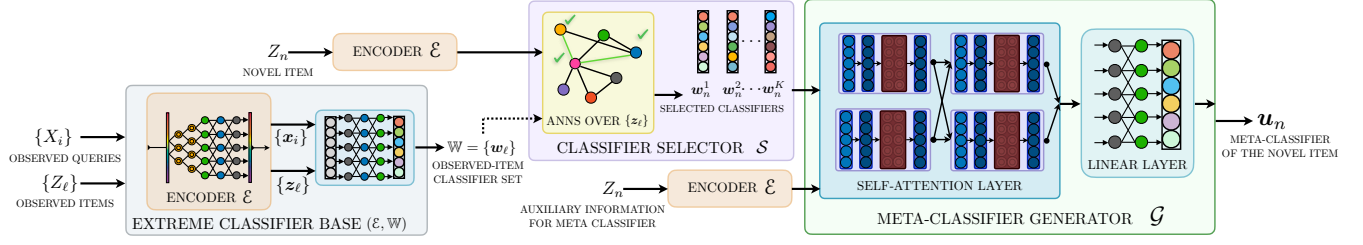
Figure 2: The IRENE extreme meta-classifier. IRENE comprises (a) A base extreme classifier encoder and the classifiers trained via a standard algorithm; (b) A classifier selector $\mathcal{S}$, which, given the encoder representation of a novel item Z_n, retrieves K classifiers based on an approximate nearest neighbor search (ANNS), and (c) A transformer-based meta-classifier generator $\mathcal{G}$. The meta-classifier u_n of Z_n is computed as given in Equation equation 1.

To facilitate training, the loss is approximated using standard negative mining strategies, with the positive, and negative-mined items associated with a query represented by $\mathbb{Z}_{o,i}^+$ and $\mathbb{Z}_{o,i}^-$, respectively. The modified loss is then given by

$$\mathcal{L}_{\mathcal{G}}(\phi) = \sum_{x_i \in \mathbb{X}} \sum_{z_\ell \in \mathbb{Z}_{s,i}} \left(C y_{i\ell} \, ln \left(x_i^\top u_\ell \right) + (1 - y_{i\ell}) \ln \left(1 - x_i^\top u_\ell \right) \right), \quad (3)$$

where $\mathbb{Z}_{o,i} = \mathbb{Z}_{o,i}^+ \cup \mathbb{Z}_{o,i}^-$ includes both positive items and mined negative items associated with query x_i.

Inference: An Approximate Nearest Neighbor Search (ANNS) index $\mathcal{A}$ is constructed over the item representations to facilitate zero-shot inference. This index encompasses either solely novel item representations for zero-shot inference or a combination of observed and novel items for generalized zero-shot inference. Thus, $\mathcal{A}$ is built over $\{z_\ell \mid z_\ell \in \mathbb{Z}_T\}$, where $\mathbb{Z}_T$ represents the set of items at inference, either $\mathbb{Z}$ or $\mathbb{Z} \cup \mathbb{Z}_n$.

At the time of inference, the embedding of a query T is calculated as $x_T = \mathcal{E}(X_T) \in \mathbb{R}^d$ with X_T being the query text. To retrieve relevant items for x_T, $\mathcal{A}$ is queried using x_T. The computational complexity of IRENE's inference is $\Omega(E + d \log(|\mathbb{Z}| + |\mathbb{Z}_n|))$, where E is the cost of encoder $\mathcal{E}$. IRENE is designed to seamlessly incorporate new items by calculating their representations and adding them to the ANNS index [41] as and when novel they arrive in the system. This feature is particularly advantageous for applications like sponsored search, where novel items are frequently introduced. In such scenarios, re-training the model with a large volume of items can be resource-intensive, making IRENE a viable and efficient solution.

5 THEORETICAL GUARANTEES

Prior to evaluating the experimental efficacy of IRENE, we establish theoretical guarantees for the zero-shot generalization of EMMETT, with a specific focus on the case of IRENE. To simplify our analysis, we represent data samples as query-item pairs, $s = (x, z)$. The dataset $\mathbb{S} = \{s\}$ is formed through two potential approaches. The most straightforward method involves randomly selecting N queries and items from the respective distributions $\mathcal{X}$ and $\mathcal{Z}$, pairing them to create $\tilde{\mathbb{S}} = \{s_i = (x_i, z_i) \mid i = 1, 2, \ldots N\}$. Alternatively, considering all possible query-item pairings from $\mathbb{X}$ and $\mathbb{Z}$ (as defined in Section 3.1), results in $\mathbb{S} = \{(x_i, z_\ell) \mid i = 1, 2, \ldots N, \ell = 1, 2, \ldots L\}$, meaning, s is drawn from the Cartesian product space $\mathcal{X} \times \mathcal{Z}$. The dataset is indexed by $j = L(i - 1) + \ell$, $j = 1, 2, \ldots M = NL$. In

this context, the extreme (meta) classification problem becomes binary, with target values y in $\mathcal{Y}$, where $y_j = 1$ indicates a positive association between item z_ℓ and the query x_i. Within the XC setting, it is well known that negatively associated pairs are significantly more likely to occur than positive ones. We assume that any set $\mathbb{S}$ contains at most κ positively associated pairs. We also assume norm bounds on the encoder representations of the queries, and the learnt classifiers, i.e., $\max_{x \in \mathcal{E}(\mathcal{X})} \|x\|_2 \leq B$ and $\max_{w \in \mathbb{W}} \|w\|_2 \leq W$, respectively.

In assessing the generalization performance of EMMETT, we recall the concepts of empirical and true risks. Given a function $f(\cdot)$, and a chosen loss function, the empirical risk over a dataset $\mathbb{S}$, and the true risk are defined as:

$$\hat{\mathcal{R}} = \frac{1}{M} \sum_{\substack{j=1 \\ s_j \sim \mathbb{S}}}^{M} \text{loss}(f(s_j), y_j) \text{ and}$$

$$\mathcal{R} = \mathbb{E}_{s \sim \mathcal{X} \times \mathcal{Z}} \left[\text{loss}(f(s), y) \right],$$

respectively. For our analysis, we use the weighted binary cross-entropy loss, $C y \ln(f) + (1 - y) \ln(1 - f)$. We assume that the function f belongs to the class $\mathcal{F}$.

The generalizability of a model is indicated by the difference between empirical and true risks. This deviation depends on both data and the model. The model's influence is typically quantified using the *Rademacher complexity* $\mathfrak{R}$ [30] of the function class $\mathcal{F}$. This complexity can be evaluated empirically over the dataset (known as the empirical Rademacher complexity $\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F})$), or across all datasets of size M (given by $\mathfrak{R}_M(\mathcal{F}) = \mathbb{E}_{\mathbb{S}} \left[\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}) \right]$. Data dependence is often highlighted by bounding the risk deviation using the McDiarmid inequality [30]. The following theorem provides a bound on the deviation of the true risk from the empirical risk in the context of meta-classification.

THEOREM 1. **(Generalization performance of EMMETT)** *Let R and $\hat{R}$ denote the true and empirical risk, respectively, and $\hat{\mathfrak{R}}_{\mathbb{S}}$ denote the empirical Rademacher complexity over set $\mathbb{S}$. Let $p \ll 1$ be the probability that a query-item pair is positively associated (i.e., $y_j = 1$), and q denote the probability that a set $\mathbb{S}$, $|\mathbb{S}| = M$ has at most κ positive pairs. Then, with probability at least $1 - \delta, \delta \in (2q, 1)$,*

we have the following generalization bound:

$$R \le \hat{R} + \hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}) + 3 \left(q + \sqrt{\frac{\ln\left(\frac{2}{\delta - 2q}\right)}{2M}} \right), \; where \quad (4)$$

$$q \le \exp\left\{-2M\left(1 - p - \frac{\kappa}{M}\right)^2\right\}. \quad (5)$$

The proof is provided in Appendix A. Theorem 1 sheds light on the generalizability of new items in XC meta classifiers. Consistent with standard findings, we note that generalization gap is inversely related to the dataset size, M. Furthermore, the choice of dataset $\mathbb{S}$ over $\tilde{\mathbb{S}}$ reveals distinct advantages when considering the probabilities p and q. Specifically, a 1-vs-all setting, wherein we consider the Cartesian product space of queries and items, will lead to better overall performance. Further, for sufficiently small κ and sufficiently large M, we have $q \approx \exp\{-2M\}$. We also observe a direct correlation between generalization, and the complexity of the function class $\mathcal{F}$. In scenarios where $\mathcal{F}$ corresponds to the class of meta-classifier generators, the associated Rademacher complexity is detailed in the following lemma:

LEMMA 2. (**Rademacher complexity of the IRENE generator**) *Let $\mathcal{F}$ be the class of functions defined in the IRENE algorithm, comprising pre-determined encoder representations and classifiers, a given classifier selector that outputs K classifiers, and $\mathcal{G}$, the meta-classifier generator. Then, the Rademacher complexity of $\mathcal{F}$ can be bounded as follows:*

$$\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}) \le O\left(B\|\mathbf{M}\|_2 \sqrt{d\ln(K+1)}\right), \quad (6)$$

where $x \in \mathbb{R}^d$ and $\mathbf{M} \in \mathbb{R}^{d \times 1}$ is the weight matrix associated with the linear layer.

The proof is provided in Appendix A. Lemma 2 yields several critical insights. Firstly, for a given XC base (consisting of an encoder and classifiers), the complexity of the meta-classifier increases logarithmically with K. This creates a balancing act: using a single classifier may result in inadequate model capacity, but increasing K increases model complexity, potentially degrading performance, and slowing down training. Ablation experiments presented in Section 7 validate this claim. Secondly, the complexity of the meta-classifier generator is independent of L, indicating that the model effectively scales in generalizing to novel labels. This scalability is attributed to the classifiers being fixed during the generator's training. The subsequent Corollary discusses the Rademacher complexity of the function class $\mathcal{F}$ when the classifiers are updated concurrently with the generator.

COROLLARY 3. (**Rademacher complexity of the IRENE generator with trainable classifier**) *Let $\mathcal{F}$ be the class of functions defined in the IRENE algorithm as in Lemma 2. Let the classifier set $\mathbb{W}$ be trainable over the meta-classifier loss. Then, the Rademacher complexity of $\mathcal{F}$ can be bounded as follows:*

$$\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}) \le O\left(B^2 W \sqrt{\frac{L}{M}} \|\mathbf{M}\|_2 \sqrt{d\ln(K+1)}\right). \quad (7)$$

This result is a consequence of Lemma 2, and an extension of Theorem 3 present in Awasthi et al. [5]. Training classifiers via the meta-classifier loss incurs a complexity in the order of $\sqrt{L}$. For the sake of completeness, we present the extension of Theorem 3 present in Awasthi et al. [5] to the XC setting, wherein the model is trained on N data points $\tilde{\mathbb{X}}$, with the loss is defined over classifiers $\mathbb{W}$. The following Lemma bounds the Rademcacher complexity of the XC classifier class:

LEMMA 4. (**Rademacher complexity of the XC Classifiers**) *(extension of Awasthi et al. [5], Theorem 3) Let $\mathcal{F}$ be the class of linear classifiers defined over the seen-item set $\mathbb{Z}_s$ in the classical XC setting (cf. Section 3.1), i.e., $\mathcal{F} = \{\langle x, w_\ell \rangle \mid \ell = 1, 2, \dots, L\}$. Then, the Rademacher complexity of $\mathcal{F}$ can be bounded as $\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}) \le \frac{LBW}{\sqrt{N}}$, where $|\mathbb{X}| = N$ is the Cardinality of the training set.*

The proof is provided in Appendix A. As expected, the complexity of the XC classifier, particularly in terms of generalizing to unseen labels, increases linearly with the number of labels. This is because the classifier's learning is dependent on the query-item pairs available during training. Therefore, as novel items are introduced, additional data becomes necessary for accurately learning the classifiers.

These findings underscore the effectiveness of the meta-classifier-based EMMETT framework in achieving zero-shot generalization. They also validate the design choices implemented in the IRENE generator and algorithm, which contribute to a model with favorable model complexity. We now proceed to provide experimental evidence to support the IRENE algorithm, complemented by ablation studies. These studies elucidate our design decisions, connecting them to the theoretical bounds established earlier.

6 EXPERIMENTAL RESULTS

Datasets: We validate the performance of IRENE across a diverse set of datasets spanning multiple applications, and label space sizes (Table 1). For LF-AmazonTitles-1.3M and LF-Wikipedia-500K, we use the experimental setup reported in Bhatia et al. [8] and for LF-AOL-270K and LF-WikiHierarchy-550K, we refer to [4]. We create the zero-shot versions of the dataset by randomly partitioning the set of items into observed and novel using a 90-10 split ratio, similar to [40]. For further details on dataset creation and statistics please refer to Appendix B in the supplementary material.

Table 1: Datasets and applications used for IRENE evaluation.

Dataset	Application Type	Feature Type
LF-AOL-270K	Query Completion	Short-Text
LF-Wikipedia-500K	Category Annotation	Long-Text
LF-WikiHierarchy-550K	Taxonomy Completion	Short-Text
LF-AmazonTitles-1.3M	Product Recommendation	Short-Text
KeywordPrediction-10M	Sponsored Search	Short-Text

Baselines: We demonstrate IRENE's efficacy, by comparing its performance when built atop leading dense retrieval encoders such as NGAME [13], ANCE [46], MACLR [47], and DPR [24]. We also compare against competitive zero-shot XC methods such as SemSup-XC [2], ZestXML [17], and MACLR.

Evaluation setting and metrics: Following [2, 17] we consider two settings: (i): zero-shot retrieval on novel items and (ii): generalized zero-shot retrieval on the combined set of observed and novel items. We report performance on metrics such as Precision@k (P@k), and Recall@k (R@k) at different truncation levels k.

Results on Benchmark datasets: Table 3 shows the zero-shot and generalized zero-shot performance of IRENE. When added to different encoders, IRENE consistently improves the performance in both the zero-shot (+1.5-40% in P@1 and +1-45% in R@10) and generalized evaluation settings (+1-29% in P@1 and +1-15% in R@10). IRENE shows larger gains on datasets such as LF-AOL-270K and LF-WikiHierarchy-550K where queries and their associated items have higher lexical dissimilarity. While modelling such relations might be difficult for small encoders, the encoder can still function as an effective neighbour selector, thereby identifying relevant classifiers and combining them via IRENE's meta classifier generator. This is particularly visible in the case of MACLR, where IRENE improves P@1 by 40.2% in zero-shot on LF-WikiHierarchy-550K. IRENE's consistent gains over a wide range of applications on different encoders show the value of information introduced by classifiers and emphasize the versatility of our approach. When compared to zero-shot XC methods like SemSup-XC, IRENE attains higher P@1 in zero-shot setting (+10.31%) while being approx. 300 times more efficient as compared to SemSup-XC (ref. Table 2) at inference. These gains across the board establish the modularity of IRENE which can be used as a plug-and-play module on top any dense retriever to obtain more accurate representation for novel items for diverse recommendation applications.

Case study on Sponsored Search: In the application of sponsored search, accurately matching user queries to billions of advertiser bid keywords presents a formidable challenge, exacerbated by varying bid amounts based on relevance and semantic relationship of the match. We demonstrate IRENE's efficacy by conducting offline experiments and online A/B tests on live search engine traffic. IRENE trained on the KeywordPrediciton-10M dataset was used to obtain representations for 100M novel keywords which were found to be 4% more accurate than those obtained from leading dense retrievers in production in R@100 metric. In live deployment, IRENE was found to increase the click-through rate (ad clicks obtained per unit query) and decrease the quick-back rate (fraction of users who quickly closed the ad) by 4.2% and 0.9% respectively. IRENE could encode a novel keyword in under 1ms and increase the percentage of good predictions by 13% as ascertained by expert judges. Please refer to Section D in Supplementary material for detailed analyses on IRENE's application to sponsored search.

IRENE's Computational Cost: Table 2 compares the time taken for generating representations for a novel item and retrieving the relevant items given a query. When incorporating a new item, NGAME performs a forward pass over a 6-layer Distil-BERT. DEXA incurs an additional cost of an ANNS search over the cluster centroids. IRENE, on the other hand, incurs additional costs from the classifier selector $\mathcal{S}$ and meta-classifier generator $\mathcal{G}$, which involves an ANNS search over the set of observed items and a forward pass through a one-layer encoder respectively. The representation generation step was executed on a single NVIDIA Tesla V100 GPU for each method and ANNS search on a 96-core CPU machine.

While being efficient at inference, IRENE's classifier selector ($\mathcal{S}$) and meta-classifier generator ($\mathcal{G}$) incur minimal training costs. On a single NVIDIA Tesla V100 GPU, the training time for the NGAME encoder for LF-AmazonTitles-1.3M-10 was 83 hours and training IRENE on top of the NGAME encoder took only 6 hours.

Table 2: Comparison of inference time (in ms) on a single V100 GPU on LF-AmazonTitles-1.3M-10. (i) generating the representation of an novel item (representation time) and (ii) Retrieving relevant items for a given query

Method	Rep. Time (per item)	Retrieval Time (per query)
NGAME	0.08	0.43
SemSup-XC	N/A	151.51
DEXA	0.48	0.43
NGAME + IRENE	0.54	0.43

7 ABLATIONS

Experiments were performed to evaluate the impact of various design choices within the components of the meta-classifier generator ($\mathcal{G}$) and the classifier selector ($\mathcal{S}$) in IRENE.

Meta-Classifier Generator $\mathcal{G}$ and classifier selector $\mathcal{S}$: Table 4 shows the effect of increasing the number of layers in IRENE's meta-classifier generator block ($\mathcal{G}$). Increasing the number of layers from 1 to 2 improves zero-shot P@1 by 1.81%. However, further increasing $\mathcal{G}$'s depth to 4 leads to a marginal decrease in performance, which could be attributed to overfitting associated with increased complexity of $\mathcal{G}$. Additionally, we try out simpler alternatives, namely sum and weighted sum, as detailed in supplementary section C. IRENE performs better than these simpler alternatives by 14% P@1 in zero-shot evaluation underscoring the significance of IRENE's attention-based meta-classifier generator.

Table 4 also shows the effect of changing K, the number of observed items retrieved by Classifier selector $\mathcal{S}$, on zero-shot performance. Increasing K from 3 to 6 results in a performance decline of approximately 3% in P@1. Further increasing K to 20 continues to decrease performance.

Input Components to Meta-Classifier Generator $\mathcal{G}$: As shown in table 14 in supplementary, keeping the classifier selector $\mathcal{S}$ and Meta-Classifier Generator $\mathcal{G}$ architecture fixed, the inputs passed to $\mathcal{G}$ were changed to encoder representations z of the selected observed items, instead of their classifiers w_ℓ. IRENE with classifiers w_ℓ of the selected observed items yields superior performance, with a significant margin of 4% on P@1. This underscores the value of leveraging high-capacity classifiers of the selected items, which contain beneficial information for representing a novel item.

Varying ratio of novel items: We observe a significant decline in the performance of NGAME by 6% as we vary the number of novel items in the LF-WikiHierarchy-550K dataset from 10% to 40%. In contrast, NGAME+IRENE displays a more resilient performance, experiencing a modest decrease of 1.5%. Additional details can be found in Table 5.

Adapting to Few-Shot Scenarios - IRENE-OneShot : A plethora of algorithms have been proposed to address few-shot scenarios, with a particular emphasis on mitigating catastrophic learning. We study the extreme case of this where only a single ground truth query is revealed for an item. IRENE seamlessly extends its capabilities to utilize the additional data available for such one-shot items, enhancing its classifier selector $\mathcal{S}$. IRENE leverages the revealed query of the one-shot item to fetch the nearest classifiers alongside the item itself. A max-voting strategy is then employed to improve the selection of observed classifiers over scenarios where only the

Table 3: Comparison of Zero-Shot and Generalized Zero-Shot Accuracies of IRENE when added to different encoders. On average IRENE improves P@1 by 9% and R@10 by 10.1% in the zero-shot setting. In the generalized setting IRENE boosts P@1 by 14.9% and R@10 by 9.8%. Cases where applying IRENE atop a base encoder improves it are boldfaced

Model	LF-AOL-270K-10				LF-WikiHierarchy-550K-10				LF-AmazonTitles-1.3M-10				LF-Wikipedia-500K-10			
	Zero-Shot		Generalized		Zero-Shot		Generalized		Zero-Shot		Generalized		Zero-Shot		Generalized	
	P@1	R@10	P@1	R@10	P@1	R@10	P@1	R@10	P@1	R@10	P@1	R@10	P@1	R@10	P@1	R@10
NGAME	30.90	54.20	20.16	38.27	46.01	58.66	66.19	27.08	36.44	15.38	45.14	30.25	46.96	65.27	81.86	69.58
NGAME+IRENE	**36.33**	**60.55**	**35.27**	**52.09**	**60.93**	**74.21**	**87.50**	**36.31**	**39.03**	**16.41**	**45.81**	**31.04**	**47.01**	**67.03**	81.79	**70.75**
ANCE	33.43	67.84	22.63	49.72	43.06	56.28	68.76	25.89	30.72	12.02	27.65	17.31	30.67	58.91	42.91	43.39
ANCE+IRENE	**36.93**	**68.12**	**30.95**	**51.96**	**48.67**	**61.01**	**84.14**	**31.53**	**32.34**	**12.61**	**29.33**	**20.65**	**44.15**	**67.39**	**67.68**	**56.66**
MACLR	11.31	18.24	9.26	7.52	30.37	35.47	59.44	14.31	21.93	28.59	**27.50**	15.99	39.56	68.53	46.59	46.62
MACLR+IRENE	**35.79**	**65.02**	**30.79**	**45.94**	**70.55**	**82.63**	**88.29**	**38.69**	**25.05**	**32.97**	26.81	**18.76**	**42.91**	**71.01**	**70.46**	**60.66**
DPR	30.38	53.82	19.71	37.99	44.84	59.29	65.19	26.73	40.98	16.82	38.18	26.93	42.90	71.20	51.54	61.84
DPR+IRENE	**36.48**	**60.76**	**35.24**	**52.08**	**61.33**	**75.21**	**87.78**	**36.46**	40.97	**16.89**	**40.88**	**28.20**	**46.68**	71.18	**74.98**	**67.19**
TF-IDF	13.74	28.05	6.61	9.90	22.79	21.44	64.50	10.88	24.15	13.30	16.33	20.40	11.53	23.89	15.07	14.79
Zest-XML	9.34	25.91	26.34	26.57	13.97	17.48	68.86	22.13	5.58	6.87	41.36	22.87	2.62	14.73	60.16	45.15
SemSup-XC	26.27	36.31	26.12	23.92	57.45	46.81	90.51	28.37	11.68	20.04	25.13	15.21	46.60	57.08	54.20	38.08
DEXA	21.68	41.85	25.09	46.76	54.83	66.89	76.18	36.17	35.19	14.66	48.19	30.89	42.76	67.37	67.98	65.86

Table 4: Ablation study on meta-classifier generator $\mathcal{G}$ and classifier selector $\mathcal{S}$ component in NGAME + IRENE on LF-WikiHierarchy-550K-10 dataset for Zero-Shot evaluation

Method	P@1	P@5	R@10
IRENE ($\mathcal{D} = 1, K = 3$)	60.93	34.67	74.21
$\mathcal{G}$ ($\mathcal{D} = 2$)	62.74	35.08	74.35
$\mathcal{G}$ ($\mathcal{D} = 4$)	61.17	34.26	73.05
$\mathcal{G}$ as Sum	45.49	25.46	59.01
$\mathcal{G}$ as wt. Sum	46.39	25.88	59.29
$K = 1$	60.96	34.56	73.99
$K = 2$	61.15	34.88	74.75
$K = 6$	57.61	32.40	71.30
$K = 20$	56.93	31.78	69.95

Table 5: Illustration of the performance variation of NGAME and NGAME+IRENE as the ratio of novel items in the WikiHierarchy dataset is adjusted from 10% to 40%. Notably, NGAME exhibits a significant decrease while NGAME+IRENE is relatively more resilient

Novel Ratio	NGAME			NGAME+IRENE		
	P@1	P@5	R@10	P@1	P@5	R@10
10	66.19	59.25	27.08	87.78	80.11	36.31
20	65.29	58.22	26.66	87.88	80.36	36.01
30	63.59	57.12	26.09	87.49	79.64	35.12
40	60.12	53.47	24.45	87.44	78.55	33.87

Table 6: Results of IRENE-OneShot, an extension of IRENE applied for one-shot retrieval. One randomly selected query is made available to all algorithms for the novel items. Evaluation is done only on the one-shot items. Here IRENE is trained on top of NGAME Encoder

Method	LF-AmazonTitles-1.3M-10			LF-Wikipedia-500K-10		
	P@1	P@5	R@10	P@1	P@5	R@10
NGAME-retrained	30.55	13.39	36.48	27.37	15.33	66.37
SemSup-XC-OneShot	13.96	6.98	19.76	46.25	13.77	57.04
IRENE-OneShot	31.47	16.41	39.03	47.01	15.48	67.03

on the revealed data, IRENE-OneShot was at least 10% better in R@10. This highlights the effectiveness of IRENE-OneShot in few-shot scenarios, showcasing its capability to surpass models that fine-tune their encoders with revealed data, all the while preserving the base encoder. This helps to reduce latency and complexity for utilizing IRENE for few-shot scenarios. Please refer to table 6 for details.

8 CONCLUSIONS

This paper studied the problem of large-scale zero-shot retrieval and developed techniques to efficiently and accurately represent novel items. It proposed a new algorithmic framework, EMMETT, for learning accurate meta-classifiers for novel items. EMMETT is a generic framework where different architectural choices and the training strategy can potentially yield algorithms with varying accuracy and efficiency trade-offs, which also presents ample opportunities for future research explorations. This paper also developed a new algorithm, IRENE, that is simple, practically deployable and can significantly boost the performance of any Siamese encoder with minimal overheads. Finally, it developed a novel theoretical framework for analyzing the generalization performance in large-scale zero-short retrieval. Comprehensive empirical validation and online A/B tests in Sponsored Search application on a major search engine demonstrated the utility of IRENE and EMMETT.

item text is considered for this purpose. This variant is denoted as IRENE-OneShot. Remarkably, IRENE-OneShot can exploit the revealed data to improve the prediction accuracy without having to fine-tune the base model. To provide a comprehensive defense against catastrophic forgetting, we explore a theoretical baseline where NGAME is trained on a consolidated dataset comprising the initial training data for observed items and the data made available for the one-shot items. Even when compared to this theoretical and optimal baseline, IRENE-OneShot was found to have 1% superior P@1. Further, compared to Semsup-XC which finetunes its encoder

REFERENCES

[1] G. Aggarwal, J. Feldman, and S. Muthukrishnan. 2006. Bidding to the top: VCG and equilibria of position-based auctions. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*.

[2] P. Aggarwal, A. Deshpande, and K. Narasimhan. 2023. SemSup-XC: Semantic Supervision for Zero and Few-shot Extreme Classification. In *ICML*.

[3] R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma. 2013. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *WWW*.

[4] Anonymous. 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

[5] P. Awasthi, N. Frank, and M. Mohri. 2020. Adversarial Learning Guarantees for Linear Hypotheses and Neural Networks. In *Proceedings of the 37th International Conference on Machine Learning*, Vol. 119. 431–441. https://proceedings.mlr.press/v119/awasthi20a.html

[6] R. Babbar and B. Schölkopf. 2017. DiSMEC: Distributed Sparse Machines for Extreme Multi-label Classification. In *WSDM*.

[7] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, M. Rosenberg, X. Song, A. Stoica, S. Tiwary, and T. Wang. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. arXiv:1611.09268 [cs.CL]

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

[9] A. Z. Broder, P. Ciccolo, M. Fontoura, E. Gabrilovich, V. Josifovski, and L. Riedel. 2008. Search Advertising Using Web Relevance Feedback. In *CIKM*.

[10] Wei-Cheng Chang, Daniel Jiang, Hsiang-Fu Yu, Choon Hui Teo, Jiong Zhang, Kai Zhong, Kedarnath Kolluri, Qie Hu, Nikhil Shandilya, Vyacheslav Ievgrafov, Japinder Singh, and Inderjit S. Dhillon. 2021. Extreme Multi-label Learning for Semantic Matching in Product Search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2643–2651.

[11] Richard Combes. 2023. An extension of McDiarmid's inequality. arXiv:1511.05240 [cs.LG]

[12] K. Dahiya, A. Agarwal, D. Saini, K. Gururaj, J. Jiao, A. Singh, S. Agarwal, P. Kar, and M. Varma. 2021. SiameseXML: Siamese Networks meet Extreme Classifiers with 100M Labels. In *ICML*.

[13] K. Dahiya, N. Gupta, D. Saini, A. Soni, Y. Wang, K. Dave, J. Jiao, K. Gururaj, P. Dey, A. Singh, D. Hada, V. Jain, B. Paliwal, A. Mittal, S. Mehta, R. Ramjee, S. Agarwal, P. Kar, and M. Varma. 2023. NGAME: Negative mining-aware mini-batching for extreme classification. In *WSDM*.

[14] K. Dahiya, D. Saini, A. Mittal, A. Shaw, K. Dave, A. Soni, H. Jain, S. Agarwal, and M. Varma. 2021. DeepXML: A Deep Extreme Multi-Label Learning Framework Applied to Short Text Documents. In *WSDM*.

[15] 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*.

[16] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise Zero-Shot Dense Retrieval without Relevance Labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). 1762–1777.

[17] N. Gupta, S. Bohra, Y. Prabhu, S. Purohit, and M. Varma. 2021. Generalized Zero-Shot Extreme Multi-label Learning. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

[18] N. Gupta, P. H. Chen, H.-F. Yu, Cho-J. Hsieh, and I. S. Dhillon. 2022. ELIAS: End-to-End Learning to Index and Search in Large Output Spaces. In *NeurIPS*.

[19] H. Jain, V. Balasubramanian, B. Chunduri, and M. Varma. 2019. Slice: Scalable Linear Extreme Classifiers trained on 100 Million Labels for Related Searches. In *WSDM*.

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

[21] Suhas Jayaram Subramanya, Fnu Devvrit, Harsha Vardhan Simhadri, Ravishankar Krishnaswamy, and Rohan Kadekodi. 2019. Diskann: Fast accurate billion-point nearest neighbor search on a single node. *Advances in Neural Information Processing Systems* 32 (2019).

[22] T. Jiang, D. Wang, L. Sun, H. Yang, Z. Zhao, and F. Zhuang. 2021. LightXML: Transformer with Dynamic Negative Sampling for High-Performance Extreme Multi-label Text Classification. In *AAAI*.

[23] Karen Spärck Jones. 2021. A statistical interpretation of term specificity and its application in retrieval. *J. Documentation* 60 (2021), 493–502. https://api.semanticscholar.org/CorpusID:2996187

[24] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-T. Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *EMNLP*.

[25] S. Khandagale, H. Xiao, and R. Babbar. 2020. Bonsai: diverse and shallow trees for extreme multi-label classification. *ML* (2020).

[26] S. Kharbanda, A. Banerjee, E. Schultheis, and R. Babbar. 2022. CascadeXML: Rethinking Transformers for End-to-end Multi-resolution Training in Extreme Multi-label Classification. In *NeurIPS*.

[27] O. Khattab and M. Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *SIGIR*.

[28] H. Kim, G. Papamakarios, and A. Mnih. 2021. The Lipschitz Constant of Self-Attention. In *Proceedings of the 38th International Conference on Machine Learning*, Vol. 139. 5562–5571. https://proceedings.mlr.press/v139/kim21i.html

[29] Y. Liu, X. Gao, and L. Gao, Q. Han. J. Shao. 2020. Label-activating framework for zero-shot learning. In *Neural Networks*, Vol. 121. 1–9.

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

[31] T. K. R. Medini, Q. Huang, Y. Wang, V. Mohan, and A. Shrivastava. 2019. Extreme Classification in Log Memory using Count-Min Sketch: A Case Study of Amazon Search with 50M Products. In *NeurIPS*.

[32] T. Mensink, E. Gavves, and C. G. M. Snoek. 2014. COSTA: Co-Occurrence Statistics for Zero-Shot Classification. In *CVPR*.

[33] A. Mittal, N. Sachdeva, S. Agrawal, S. Agarwal, P. Kar, and M. Varma. 2021. ECLARE: Extreme Classification with Label Graph Correlations. In *WWW*.

[34] Y. Qu, Y. Ding, J. Liu, K. Liu, R. Ren, W. X. Zhao, D. Dong, H. Wu, and H. Wang. 2021. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering.

[35] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. 3, 4 (apr 2009), 333–389. https://doi.org/10.1561/1500000019

[36] Bernardino Romera-P. and P. Torr. 2015. An embarrassingly simple approach to zero-shot learning. In *ICML*.

[37] P. Rusmevichientong, D. P. Williamson, and D. B. Shmoys. 2006. An optimization framework for finding revenue maximizing bid prices in keyword auctions. In *WWW*.

[38] D. Saini, A.K. Jain, K. Dave, J. Jiao, A. Singh, R. Zhang, and M. Varma. 2021. GalaXC: Graph Neural Networks with Labelwise Attention for Extreme Classification. In *WWW*.

[39] Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Tianyi Zhou, and Daxin Jiang. 2023. Large Language Models are Strong Zero-Shot Retriever. arXiv:2304.14233

[40] Daniel Simig, Fabio Petroni, Pouya Yanki, Kashyap Popat, Christina Du, Sebastian Riedel, and Majid Yazdani. 2022. Open Vocabulary Extreme Classification Using Generative Models. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics, 1561–1583. https://doi.org/10.18653/v1/2022.findings-acl.123

[41] Aditi Singh, Suhas Jayaram Subramanya, Ravishankar Krishnaswamy, and Harsha Vardhan Simhadri. 2021. FreshDiskANN: A Fast and Accurate Graph-Based ANN Index for Streaming Similarity Search. arXiv:2105.09613 [cs.IR]

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[43] J. Vuckovic, A. Baratin, and R. Tachet des Combes. 2020. A Mathematical Theory of Attention. arXiv:2007.02876 [stat.ML]

[44] Y. Wang, J. Liu, Y. Wang, C. Tai, J. Shao, J. Ma, and C. Zhai. 2015. A noise-filtered under-sampling scheme for imbalanced classification. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*.

[45] Ji Xin, Chenyan Xiong, Ashwin Srinivasan, Ankita Sharma, Damien Jose, and Paul Bennett. 2022. Zero-Shot Dense Retrieval with Momentum Adversarial Domain Invariant Representations. In *Findings of the Association for Computational Linguistics: ACL 2022*. 4008–4020.

[46] L. Xiong, C. Xiong, Y. Li, K.-F. Tang, J. Liu, P. Bennett, J. Ahmed, and A. Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *ICLR*.

[47] Y. Xiong, W.-C. Chang, C.-J. Hsieh, H.-F. Yu, and I. Dhillon. 2021. Extreme Zero-Shot Learning for Extreme Text Classification. arXiv:2112.08652 [cs.LG]

[48] C. Yun, S. Bhojanapalli, A. Singh Rawat, S. Reddi, and S. Kumar. 2020. Are Transformers universal approximators of sequence-to-sequence functions?. In *International Conference on Learning Representations*. https://openreview.net/forum?id=ByxRM0Ntvr

[49] J. Zhang, W. C. Chang, H. F. Yu, and I. Dhillon. 2021. Fast multi-resolution transformer fine-tuning for extreme multi-label text classification. In *NeurIPS*.

[50] Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2023. Dense Text Retrieval based on Pretrained Language Models: A Survey. *ACM Trans. Inf. Syst.* (2023).

9 ETHICAL CONSIDERATIONS

Our data usage and service provision align with the approvals of our legal and ethical committees. Sponsored search plays a vital role in guaranteeing free and convenient access to information for billions of users worldwide, thereby sustaining free web engines. Our research contributes to the enhancement of the online search experience, offering accessible information to millions of users around the world while enabling advertisers to connect with relevant audiences. This mutual relationship cultivates a more effective and valuable advertising ecosystem, benefiting both users and advertisers. Furthermore, our endeavors support the financial growth of small and medium businesses, including local establishments, by increasing revenue, expanding market reach, and reducing costs associated with reaching new customers.

A   PROOFS OF THEOREMS

We now present the proofs of the various theorems presented in the main manuscript. We recall the definitions presented in Section 5. The data can be represented as query-item pairs, given by $s = (x, z)$. A key design choice is on forming the dataset $\mathbb{S} = \{s\}$. The simplest formulation is that of selecting N queries and items randomly from the underlying distributions $\mathcal{X}$ and $\mathcal{Z}$ respectively, and pairing them, to form the dataset $\tilde{\mathbb{S}} = \{s_i = (x_i, z_i) \mid i = 1, 2, \dots N\}$. Alternatively, given $\mathbb{X}$ and $\mathbb{Z}$, defined as $\mathbb{X} = \{x_i\}_{i=1}^N$, and $\mathbb{Z}_s = \{z_\ell\}_{\ell=1}^L$ respectively, we can consider all possible pairing of queries and items, giving us $\mathbb{S} = \{(x_i, z_\ell) \mid i = 1, 2, \dots N, \ell = 1, 2, \dots L\}$, i.e., s is drawn from the Cartesian product space $\mathcal{S} = \mathcal{X} \times \mathcal{Z}$. Note that $\mathbb{S}$ can be indexed as $j = L(i - 1) + \ell$, $j = 1, 2, \dots M = NL$. The extreme (meta) classification problem is now one of binary classification, with targets y drawn from the space $\mathcal{Y}$, that satisfy $y_j = 1$ if item z_ℓ is positively associated with the query x_i. In the XC setting, it is well known that negatively associated pairs are significantly more likely to occur than positive ones. We assume that in any set $\mathbb{S}$, we have at most κ positively associated pairs. Let $\max_{x \in \mathcal{E}(\mathcal{X})} \|x\|_2 \leq B$ and $\max_{w \in \mathbb{W}} \|w\|_2 \leq W$ be the bounds on the norm of the encoder representations of the queries and the learnt classifiers, respectively.

Before we proceed with the proofs, we recall a few key definitions.

Empirical and True Risk: Given a function $f(\cdot)$ drawn from a function space $\mathcal{F}$, and a loss function, the empirical risk over $\mathbb{S}$, and the true risk, are given by

$$\hat{R} = \frac{1}{M} \sum_{\substack{j=1 \\ s_j \sim \mathbb{S}}}^M \text{loss}(f(s_j), y_j) \quad \text{and} \quad R = \mathbb{E}_{s \sim \mathcal{X} \times \mathcal{Z}} \left[\text{loss}(f(s), y) \right], \quad \text{respectively.}$$

Rademacher Complexity [30]: Given a function f drawn from the function class $\mathcal{F}$, the emprical Rademacher complexity defined over the set $\mathbb{S}$, and the Rademacher complexity over all sets of size M are given by:

$$\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}) = \frac{1}{M} \mathbb{E}_{\sigma} \left[\sup_{f \in \mathcal{F}} \sum_{j=1}^M \sigma_j f(s_j) \right], \quad \text{and} \quad \mathfrak{R}_M(\mathcal{F}) = \mathbb{E}_{\mathbb{S}} \left[\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}) \right], \tag{8}$$

respectively, where $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_M)$, whose entries are independent random variables drawn from the *Rademacher distribution* i.e. $\Pr(\sigma_j = +1) = \Pr(\sigma_j = -1) = 1/2, \forall j$.

McDiarmid Inequality [11]: McDiarmid's inequality is a concentration inequality to bound the deviation of the sampled value from the expected value. We consider an extension of the standard inequality, to the case where the function under consideration Φ does not strictly satisfy the bounded differences property, but large differences remain very rare.

Let $\Phi : \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_M \to \mathbb{R}$ be a function acting on the dataset $\mathbb{S}$, with $s_j \in \mathcal{S}_j$ and $\mathcal{S}' \subseteq \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_M$ be a subset of its domain and let $c_1, c_2, \dots, c_n \geq 0$ be constants such that all pairs $(s_1, \dots, s_M) \in \mathcal{S}'$ and $(s_1', \dots, s_n') \in \mathcal{S}'$, satisfy the bounded difference property:

$$\left| \Phi(s_1, \dots, s_M) - \Phi(s_1', \dots, s_M') \right| \leq \sum_{j : s_j \neq s_j'} c_j.$$

All datasets drawn from $\mathcal{S}'$ can be viewed as *good sets* that satisfy bounded difference. Then, given a random dataset $\mathbb{S}$, with probability $q = 1 - \Pr(\mathbb{S} \in \mathcal{S}')$ (which is the probability of drawing a *bad set*), the following holds:

$$\Pr\left(\left| \Phi(\mathbb{S}) - \mathbb{E}_{\mathbb{S}} \left[\Phi(\mathbb{S}) \right] \right| \geq \epsilon \right) \leq 2q + 2 \exp \left\{ -\frac{2 \left(\max\left\{ 0, \epsilon - q \sum_j c_j \right\} \right)^2}{\sum_j c_j^2} \right\}.$$

Hoeffding's Inequality [30]: The Hoeffding's inequality is a special case of the McDiarmid inequality. Let $Y_1, Y_2, \dots Y_M$ be M independent Bernoulli-distributed random variables. Let the sum be denoted as $S_M = Y_1 + Y_2 + \dots + Y_M$. Then, $\forall t > 0$, we have

$$\Pr\left(S_M - \mathbb{E}_Y \left[S_M \right] \geq t \right) \leq \exp \left\{ -\frac{2t^2}{M} \right\}.$$

Proof of Theorem 1: Consider the empriical and true risk defined in Equation 8. The proof follows by deriving the generalization bound using the McDiarmid inequality, followed by using the Hoeffding inequality to bound the probability q. Without loss of generality, we redefine the loss to have the target labels $y_j \in \{-1, 1\}$, giving rise to the following form of the loss:

$$g(s, y) = \text{loss}\left(f(s), y \right) = \left(\frac{1 - y}{2} \right) f(s) - C \left(\frac{1 + y}{2} \right) f(s). \tag{9}$$

Let $\Phi(\mathbb{S})$ be defined as follows

$$\Phi(\mathbb{S}) = \sup_{f \in \mathcal{F}} \left\{ \mathbb{E}_s \left[g(s, y) \right] - \frac{1}{M} \sum_{j=1}^M g(s_j, y_j) \right\} = \sup_{f \in \mathcal{F}} \left\{ \mathbb{E} \left[g(s, y) \right] - \hat{\mathbb{E}} \left[g(s_j, y_j) \right] \right\}$$

11

The subsequent steps are consistent with those in deriving the Rademacher complexity in the standard setting [30], but modified to accommodate the extension of the McDiarmid inequality. Let $\mathbb{S}$ and $\mathbb{S}'$ be two sample sets, differing only in element $\boldsymbol{s}_k$. Then, we have:

$$\Phi(\mathbb{S}) - \mathbb{E}_{\mathbb{S}}\left[\Phi(\mathbb{S})\right] \leq \frac{1}{M} \sup_{f \in \mathcal{F}} \left\{ g(\boldsymbol{s}_k, y_k) - g'(\boldsymbol{s}_k, y_k) \right\} \leq c_j$$

if $\mathcal{S}$ and $\mathcal{S}'$ are drawn from the *good set* $\tilde{\mathcal{S}}$ (which we discuss shortly), then, we have $c_j = \frac{1}{M}$, $\forall j$, and McDiarmid's inequality can be applied to $\Phi(\mathbb{S})$ to obtain the following, for $\epsilon > q$:

$$\Pr\left(\left|\Phi(\mathbb{S}) - \mathbb{E}_{\mathbb{S}}\left[\Phi(\mathbb{S})\right]\right| \geq \epsilon\right) \leq 2q + 2\exp\left\{-\frac{2\left(\max\left\{0, \epsilon - q\sum_j c_j\right\}\right)^2}{\sum_j c_j^2}\right\}$$

$$= 2q + 2\exp\left\{-2M\left(\max\left\{0, \epsilon - q\right\}\right)^2\right\}$$

$$= \underbrace{2q + 2\exp\left\{-2M\left(\epsilon - q\right)^2\right\}}_{\delta}$$

Solving for ϵ on the right hand side, we obtain:

$$\delta = 2q + 2\exp\left\{-2M\left(\epsilon - q\right)^2\right\}$$

$$\Rightarrow 0 = \epsilon^2 + (-2q)\epsilon + \left(q^2 + \frac{1}{2M}\ln\left(\frac{2}{\delta - 2q}\right)\right)$$

$$\Rightarrow \epsilon = q + \sqrt{\frac{\ln\left(\frac{2}{\delta - 2q}\right)}{2M}},$$

for $\delta \in (2q, 1)$. Substituting the above into McDiarmid's inequality, we obtain, with probability $1 - \delta$,

$$\Phi(\mathbb{S}) \leq \mathbb{E}_{\mathbb{S}}\left[\Phi(\mathbb{S})\right] + \left(q + \sqrt{\frac{\ln\left(\frac{2}{\delta - 2q}\right)}{2M}}\right).$$

It is straightforward to derive the bound $\mathbb{E}_{\mathbb{S}}\left[\Phi(\mathbb{S})\right] \leq 2\mathfrak{R}_M(\text{loss} \circ \mathcal{F})$ [30]. Similarly, by applying the McDiarmid inequality to $\mathfrak{R}$, we get:

$$\mathfrak{R}_M(\text{loss} \circ \mathcal{F}) \leq \hat{\mathfrak{R}}_{\mathbb{S}}(\text{loss} \circ \mathcal{F}) + \left(q + \sqrt{\frac{\ln\left(\frac{2}{\delta - 2q}\right)}{2M}}\right).$$

Further, for the given form of the loss, we have the following result:

$$\hat{\mathfrak{R}}_{\mathbb{S}}(\text{loss} \circ \mathcal{F}) = \frac{1}{M}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{f \in \mathcal{F}} \sum_{j=1}^{M} \sigma_j(\text{loss} \circ f)(\boldsymbol{s}_j)\right]$$

$$= \frac{1}{M}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{f \in \mathcal{F}} \sum_{j=1}^{M} \sigma_j\left(\left(\frac{1 - y_j}{2}\right)f(\boldsymbol{s}_j) - C\left(\frac{1 + y_j}{2}\right)f(\boldsymbol{s}_j)\right)\right]$$

$$= \frac{1}{2M}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{f \in \mathcal{F}} \sum_{j=1}^{M} \sigma_j f(\boldsymbol{s}_j)(1 - C) - \sigma_j y_j f(\boldsymbol{s}_j)(1 + C)\right]$$

Since σ_j and $-\sigma_j y_j$ are distributed identically, we have:

$$\hat{\mathfrak{R}}_{\mathbb{S}}(\text{loss} \circ \mathcal{F}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{\sigma}}\left[2\frac{1}{M}\sup_{f \in \mathcal{F}} \sum_{j=1}^{M} \sigma_j f(\boldsymbol{s}_j)\right] = \hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F})$$

Putting the above together, and substituting in for $\Phi(\mathbb{S})$, we obtain

$$R \leq \hat{R} + \hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}) + 3\left(q + \sqrt{\frac{\ln\left(\frac{2}{\delta - 2q}\right)}{2M}}\right),$$

which completes part of the proof of Theorem 1.

What remains, is to derive q, the probability of drawing a *bad set* $\mathbb{S}$. A *good set* is one which satisfies the bounded difference condition. Consider the set $\mathbb{S} = \{s_1, s_2, \ldots, s_M\}$. Let p denote the probability that a query-item pair is positively correlated. Then, by the definition of the loss in Equation 9, for positively correlated pairs, the maximum error (over all $f \in \mathcal{F}$ is W, while for negatively correlated pairs, it is 1. For $c_j = \frac{1}{M}$, we require that the sample s_j is a negative pair. Let a *good set* $\mathbb{S}$ have at most κ positively associated pairs. Then, a *bad set* contains at least $M - \kappa$ positively associated pairs. Then, q is the probability of drawing a set $\mathbb{S}$ with at least $M - \kappa$ positively associated pairs.

To derive this probability, consider indicator variables $Y_j = \mathbb{I}_{[s_j \text{ contains a positive pairing}]}$. Let $S_M = \sum_j Y_j$ denote the sum. Then, we have $\mathbb{E}_Y [S_M] = pM$. Applying the Hoeffding inequality with $t = M - \kappa - pM$, we get:

$$\Pr\left(S_M - \mathbb{E}_Y [S_M] \geq t\right) \leq 2 \exp\left\{-\frac{2t^2}{M}\right\}$$

$$\Rightarrow \Pr\left(S_M - pM \geq M - \kappa - pM\right) = \Pr\left(S_M \geq M - \kappa\right) \leq 2 \exp\left\{-\frac{2(M - \kappa - pM)^2}{M}\right\}.$$

Simplifying, we get:

$$q = \Pr\left(\text{set } \mathbb{S} \text{ contains at least } M - \kappa \text{ positively associated pairs}\right) \leq 2 \exp\left\{-2M\left(1 - p - \frac{\kappa}{M}\right)^2\right\},$$

which is significantly smaller than 1, given large M (which is true in the case of XC, where we have large N and L) and $\kappa = 0$, which is the setting under which the required McDiarmid inequality is defined. This completes the proof of Theorem 1.

Proof of Lemma 4: We now state and discuss the proof of Lemma 4, which is an extension of Theorem 3 from Awasthi et al. [5].

LEMMA. (**Rademacher complexity of the XC Classifiers**) (*extension of Awasthi et al. [5], Theorem 3*) *Let $\mathcal{F}$ be the class of linear classifiers defined over the seen-item set $\mathbb{Z}_s$ in the classical XC setting (cf. Section 3.1), i.e., $\mathcal{F} = \{\langle x, w_\ell\rangle \mid \ell = 1, 2, \ldots, L\}$, where $x \in \mathbb{X}$. Then, the Rademacher complexity of $\mathcal{F}$ can be bounded as follows:*

$$\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}) \leq \frac{LBW}{\sqrt{N}},$$

where $|\mathbb{X}| = N$ is the Cardinality of the training set.

PROOF. Let $\mathcal{F}_\ell = \{\langle x, w_\ell\rangle \mid \|w_\ell\|_2 \leq W\}$. Awasthi et al. [5] showed that $\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}_\ell) \leq \frac{W}{N}\|\mathbf{X}\|_F$, where $\mathbf{X}$ is a matrix formed with the elements of $\mathbb{X}$. For bounded data, $\max_{x \in \mathcal{E}(\mathcal{X})} \|x\|_2 \leq B$, we have $\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}_\ell) \leq \frac{WB}{\sqrt{N}}$. Given $\mathcal{F}$ with L classifiers, the Rademacher complexity is bounded by $\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}_\ell) \leq \frac{LBW}{\sqrt{N}}$. □

Proof of Lemma 2 and Corollary 3: We now derive the bound on the Rademacher complexity of the IRENE meta-classifier generator. Recall the Lemma

LEMMA. (**Rademacher complexity of the IRENE generator**) *Let $\mathcal{F}$ be the class of functions defined in the IRENE algorithm, comprising pre-determined encoder representations and classifiers, a given classifier selector that outputs K classifiers, and $\mathcal{G}$, the meta-classifier generator. Then, the Rademacher complexity of $\mathcal{F}$ can be bounded as follows:*

$$\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}) \leq O\left(B\|\mathbf{M}\|_2 \sqrt{d \ln(K + 1)}\right),$$

where $x \in \mathbb{R}^d$ and $\mathbf{M} \in \mathbb{R}^{d \times 1}$ is the weight matrix associated with the linear layer.

PROOF. Recall the IRENE meta-classifier generator:

$$\mathcal{F} = \left\{(x_i, z_\ell) \rightarrow \left\langle x_i, \text{Linear}\left(\text{SelfAttention}\left(C_{lf}\left(\mathcal{S}(z_\ell)\right)\right)\right)\right\rangle\right\}, \quad \text{where}$$

$$\mathcal{F}_3 = C_{lf}\left(\mathcal{S}(z_\ell)\right), \quad \mathcal{F}_2 = \text{SelfAttention}(\mathcal{F}_3), \quad \text{and} \quad \mathcal{F}_1 = \text{Linear}(\mathcal{F}_2) = \mathbf{M}\mathcal{F}_2 + b$$

The proof follows by repeatedly applying Talagrand's lemma, which states that, given an L-Lipschitz continuous function g, and the function class $\mathcal{F}$, we have $\hat{\mathfrak{R}}_{\mathbb{S}}(g \circ \mathcal{F}) \leq L\,\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F})$. Applying Talagrand's lemma to $\mathcal{F}$, we get:

$$\hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}) \leq B\,\hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}_1) \tag{10}$$

$$\leq B\|\mathbf{M}\|_2\,\hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}_2) \tag{11}$$

$$\leq B\|\mathbf{M}\|_2\sqrt{d \ln (K + 1)}\,\hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}_3) \tag{12}$$

$$\leq O\left(B\|\mathbf{M}\|_2\sqrt{d \ln (K + 1)}\right), \tag{13}$$

where Equation 10 is a consequence of the boundedness of x, Equation 11 is obtained by considering the Lipschitz constant of a linear transformation layer with vector-valued weights, and Equation 12 is obtained by applying the Lipschitz constant of a self-attention layer, derived by Vuckovic et al. [43], where in turn, d is the dimensionality of the input sequence, and K is the number of classifiers selected by $\mathcal{S}$. We note that, while Kim et al. [28] provide a tighter bound in the context of L_2 multi-head attention, their result does not hold for the dot-product-based self-attention block considered above. Equation 13 is a consequence of considering a fixed set of classifiers $\mathbb{W}$, and a pre-determined classifier selector algorihtm. Therefore, these blocks merely add a constant factor to the complexity to the model $\mathcal{F}$ class, completing the proof of Lemma 2. □

The analysis can be extended to derive Corollary 3 by incorporating trainable classifiers. Recall the statement of the Corollary:

COROLLARY. (**Rademacher complexity of the IRENE generator with trainable classifier**) *Let $\mathcal{F}$ be the class of functions defined in the IRENE algorithm as in Lemma 2. Let the classifier set $\mathbb{W}$ be trainable over the meta-classifier loss. Then, the Rademacher complexity of $\mathcal{F}$ can be bounded as follows:*

$$\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}) \le O\left(B^2 W \sqrt{\frac{L}{M}} \|\mathbf{M}\|_2 \sqrt{d \ln(K+1)}\right),$$

where $x \in \mathbb{R}^d$ and $\mathbf{M} \in \mathbb{R}^{d \times 1}$ is the weight matrix associated with the linear layer.

PROOF. THe above result can be obtained by extending the proof of Lemma 2:

$$\hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}) \le B\|\mathbf{M}\|_2 \sqrt{d \ln(K+1)} \, \hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}_3)$$

$$\le O\left(B\|\mathbf{M}\|_2 \sqrt{d \ln(K+1)} \, \hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}_{\text{clf}})\right),$$

where $\hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}_{\text{clf}})$ can be obtained from Lemma 4, considering the dataset $\mathbb{S}$ with $M = NL$ samples, and L classifiers associated with the observed items, which yields the desired result:

$$\hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}) \le O\left(B\|\mathbf{M}\|_2 \sqrt{d \ln(K+1)} \frac{LW}{M} \|\mathbf{X}\|_F\right)$$

$$= O\left(\sqrt{\frac{L}{N}} B^2 W \|\mathbf{M}\|_2 \sqrt{d \ln(K+1)}\right).$$

This completes the proof of Corollary 3. □

B   DATASET CREATION AND STATISTICS

IRENE was evaluated on a diverse set of datasets spanning multiple applications like product recommendation (LF-AmazonTitles-1.3M), category annotation (LF-Wikipedia-500K), query completion (LF-AOL-270K), and taxonomy completion (LF-WikiHierarchy-600K). Additionally, IRENE was evaluated on a proprietary query-to-keyword matching dataset. Table 7 describes the statistics of the different datasets.

Zero-shot splits of the dataset were created in the following manner: Given the original dataset consisting of L items and a split fraction s, sL items were randomly selected to form the novel item set. The remaining $(1-s)L$ items along with their associated queries form the training corpus. The novel test set was created based on connections between test queries and novel items, while for the generalized evaluation, we refer to the data source from [4] or [8]. We use the notation *dataset-split* (e.g., LF-AmazonTitles-1.3M-10 for a 10% unseen ratio split) to refer to the zero-shot version of the dataset.

Table 7: Dataset Statistics.

Dataset	# Observed Qs	# novel Test Qs	# Gen Test Qs	# Observed Items	# novel Items
KeywordPrediction-10M	220,845,427	5,022,052	5,022,052	4,999,996	5,435,443
LF-AOL-270K-10	3,689,542	68,491	519,352	245,543	27,282
LF-WikiHierarchy-550K-10	1,587,567	339,086	397,870	494,733	54,970
LF-AmazonTitles-1.3M-10	2,225,354	624,830	970,237	1,174,739	130,526
LF-Wikipedia-500K-10	1,781,890	271,620	783,743	450,963	50,107

C   ABLATIONS DISCUSSIONS

In the context of generating a new item representation $\mathbf{u}_\ell$, let $\mathcal{S}(\mathbf{z}_\ell)$ denote the set of shortlisted seen items for a given item l, with $\mathbf{w}_\ell \in \mathcal{S}(\mathbf{z}_\ell)$. Here, $\mathbf{w}_\ell^i$ represents the classifier of the ith neighbor, and $\mathbf{z}_\ell$ is the encoder representation of the new item.

Summation Formulation: The summation formulation for $\mathbf{u}_\ell$ is given by aggregating the encoder representation of the new item with the classifiers of its K neighbors as follows:

$$\mathbf{u}_\ell = \left(\mathbf{z}_\ell + \mathbf{w}_\ell^1 + \mathbf{w}_\ell^2 + \ldots + \mathbf{w}_\ell^K\right) \tag{14}$$

Table 8: Results on 100M zero-shot keywords on the search engine. Method Ms are anonymized in-production dense retrievers. All algorithms are provided with just the text of a keyword to get its representation

Method	R@30	R@50	R@100	P@30	P@50	P@100
M1	24.74	32.82	48.46	39.10	36.60	33.25
M2	19.93	26.49	38.99	35.72	33.11	29.63
M3	19.11	25.84	39.33	31.48	28.93	25.69
M4	28.79	39.64	62.19	44.24	41.96	38.72
NGAME+IRENE	**30.53**	**42.00**	**66.68**	**45.64**	**43.40**	**40.11**

Table 9: Expert judges labeling results on KeywordPrediction-10M dataset

Method	% of good quality predictions
NGAME	64.06
NGAME+IRENE	73.16
NGAME+IRENE- OneShot	77.05

Weighted Summation Formulation: In the weighted summation formulation, $\mathbf{u}_\ell$ is computed by a weighted sum of the encoder representation and the classifiers of its neighbors, where the weights c^i are learned parameters:

$$\mathbf{u}_\ell = \left(c^0 \mathbf{z}_\ell + c^1 \mathbf{w}_\ell^1 + c^2 \mathbf{w}_\ell^2 + \ldots + c^K \mathbf{w}_\ell^K \right) \tag{15}$$

D DETAILED DISCUSSION ON SPONSORED SEARCH

Sponsored search is essentially a match-making system between users and advertisers with the objective to optimize user experience while searching for knowledge and help advertisers reach the right set of users who might be interested in their product/service [1]. Users encode their intent in short pieces of text called queries. Similarly, advertisers also bid on short pieces of text, relevant to their ads. One critical component of the sponsored search pipeline is the task of matching user queries to these advertiser bid keywords. Most search engines currently follow different semantics to do this matching called match-types[9]. Matching user queries to advertiser keywords is a nuanced and challenging problem as maintaining the semantics of the match-type is essential to advertisers who often bid differently on different match-types for the same keyword[37]. Furthermore, given the placement of this matching task at the forefront of the ads retrieval pipeline, any improvements in accuracy within this application can yield super-linear benefits for downstream components [44]. Please note that the consideration of a larger truncation factor (30, 50, and 100) for the Precision metric is deliberate. Retrieval algorithms like IRENE precede reranking algorithms, necessitating the prediction of a larger candidate set for input into these reranking algorithms.

Online Results IRENE underwent deployment on a prominent search engine for conducting A/B tests with live search engine traffic. Throughout the live A/B test interaction on the search engine, IRENE was systematically compared against an extensive control ensemble featuring diverse algorithms, encompassing not only DR algorithms but also prominent generative, graph-based, and IR algorithms. Evaluation of performance was based on live metrics. The findings revealed that IRENE led to a 4.2% increase in the click-through rate (ad clicks obtained per unit query) and a 0.9% decrease in the quick-back rate (fraction of users quickly leaving the ad landing page due to perceived irrelevance). These results underscore the value creation for users, indicating that IRENE effectively presented more relevant ads to the audience. Furthermore, IRENE demonstrated a noteworthy 7.8% increase in keyword density (average number of keywords surviving quality control and relevance filters), affirming the quality of its predictions. Additionally, IRENE achieved a click efficiency of 150%, signifying that for every 2% increase in ad impressions, ads selected by IRENE garnered 3% more clicks. In labeling by expert judges, IRENE was found to increase the percentage of good keyword predictions by 9% (refer to 9 for details). Notably, IRENE successfully matched queries such as "grainger" and "bitwarden" to advertiser keywords like "industrial supply" and "password manager," respectively. It is essential to highlight that these predictions, not relying on text matching, were not replicated by any in-production algorithm. Please refer to Table 10 for more such predictions made by IRENE but missed by the control ensemble.

Furthermore, we conduct a direct comparison of IRENE with prominent proprietary and public Dense Retrieval (DR) algorithms currently in production. Specifically, we randomly sample 100 million advertiser keywords introduced into the system after the period covered by the training data scraping. Additionally, we select some of the top-performing dense retrieval encoders deployed in production and pit IRENE against them in recommending keywords from this 100 million set for a sampled array of queries. For intellectual property reasons, the names of these algorithms are anonymized, and the results are detailed in Table 8. IRENE was found to be at least 4% superior than the next best dense retriever in R@100. As novel items stream into the system, it is be necessary to frequently encode the items and include them in the ANNS index. Table 2 shows that IRENE adds only minimal overhead on top of a language encoder and can get the item representation in less than one ms. Further, the integration of updatable ANNS algorithms, such as Fresh-DiskANN [41], can greatly reduce deployment time for novel items.

Offline Results To conduct offline experiments, we curated the KeywordPrediction-10M dataset by mining the logs of a commercial search engine within a specific timeframe. The dataset comprised user-typed queries and the corresponding bid keywords for surfaced

Table 10: Advertiser keywords predicted by IRENE for a user query, but overlooked by the production ensemble comprising leading dense retrieval, graph-based, XC, and generative language models. IRENE extends its capability beyond textual similarity by endowing the query representation with world knowledge obtained from classifiers of similar observed items

User Query	Advertiser Keyword
best pfmea control plan software	best cmms
work order app hydraulics	plumbing work order software
firewalla gold att fiber connection	att ethernet network
nn2 best home fumigation companies ventura oxnard	bug removal in oxnard
youtube	streaming services
netbenefits com login	fidelity retirement account
financial services crm software	sap customer management software
hypokalemia	high potassium in the blood
kaiser options	healthinsurance
23andme	genetic screening

Table 11: Results on KeywordPrediction-10M dataset.

Method	R@30	R@50	R@100	P@30	P@50	P@100
Evaluation only on novel items						
ANCE	31.37	42.87	72.45	76.3	68.40	56.32
NGAME	33.76	48.07	74.13	76.52	69.32	57.67
NGAME+IRENE	**34.79**	**49.81**	**77.32**	**78.70**	**71.76**	**60.18**
Semsup-XC	34.43	49.06	74.99	77.53	70.15	58.04
NGAME+IRENE-OneShot	**35.87**	**50.95**	**78.46**	**79.75**	**72.84**	**61.20**

advertisements, forming query-keyword training pairs. These pairs underwent basic sanity filters based on click-through rate (CTR), clicks, and impressions to generate the training dataset. Named KeywordPrediction-10M, the dataset encompassed approximately 5 million items and 220 million training queries. For additional details, refer to table 7. As presented in Table 11, IRENE demonstrated a superiority of at least 3% in Recall@100 and at least 2% in Precision@30 compared to leading Dense Retrieval (DR) algorithms NGAME and ANCE.

One-shot extension We further study the extension of IRENE to scenarios involving items that have received precisely one click, representing an exploration of IRENE's performance at the extreme tail of observed items. In this context, IRENE leverages the revealed query for a one-shot tail item to enhance its classifier selector. The revealed query of the one-shot item is utilized to retrieve the nearest classifiers, along with the item itself. A max-voting strategy is then employed to select superior observed classifiers compared to cases where only the item text is used for this purpose. In comparison to SemsupXC, which refines its language encoder with new click data, IRENE demonstrated a superiority of approximately 3% and 2% in Recall@100 and Precision@30, respectively. It's noteworthy that fine-tuning models deployed in production, as undertaken by SemsupXC, introduces latency and complexity costs, making it less desirable. Additionally, algorithms that fine-tune the trained model on revealed data, such as SemsupXC, necessitate rebuilding the Approximate Nearest Neighbors (ANNS) index from scratch. In contrast, IRENE adopts updatable ANNS algorithms, akin to many Dense Retrieval (DR) algorithms, allowing it to leverage revealed data for improved prediction accuracy without the need to fine-tune the base model. Hence, IRENE can make use of the revealed data to improve the prediction accuracy without having to fine-tune the base model. This helps to reduce latency and complexity in an online serving infrastructure.

Table 12: Zero Shot Accuracies of different encoders when combined with IRENE. Averaged across base encoders and datasets, IRENE improves P@1, P@5, and R@10 by 9%, 4.2%, and 10.1%, respectively.

Model	LF-AOL-270K-10							
	R@3	R@5	R@10	R@30	R@100	P@1	P@3	P@5
NGAME	43.90	48.8	54.2	60.14	65.39	30.90	15.43	10.32
NGAME+IRENE	49.80	55.04	60.55	66.61	71.64	36.33	17.45	11.59
ANCE	51.81	59.63	67.84	77.38	84.78	33.43	18.10	12.52
ANCE+IRENE	53.71	60.47	68.12	76.83	83.61	36.93	18.76	12.70
MACLR	13.97	15.63	18.24	21.46	30.73	11.31	4.96	3.33
MACLR+IRENE	51.27	57.80	65.02	73.35	79.75	35.79	17.92	12.14
DPR	43.44	48.53	53.82	59.71	64.84	30.38	15.26	10.24
DPR+IRENE	49.99	55.22	60.76	66.9	71.82	36.48	17.51	11.62
TF-IDF	20.10	23.76	28.05	34.10	39.13	13.74	7.27	5.08
Zest-XML	22.73	23.49	25.91	29.28	35.1	9.34	10.21	10.79
Adam	30.49	33.96	38.53	45.81	56.31	23.02	10.80	7.22
SemSup-XC	31.50	33.8	36.31	39.93	42.92	26.27	11.16	7.19
DEXA	31.64	36.50	41.85	48.81	55.59	21.68	11.16	7.73

Model	LF-WikiHierarchy-550K-10							
	R@3	R@5	R@10	R@30	R@100	P@1	P@3	P@5
NGAME	37.96	47.24	58.66	73.67	84.31	46.01	32.93	25.63
NGAME+IRENE	51.97	63.46	74.21	82.52	87.55	60.93	44.77	34.67
ANCE	35.97	44.92	56.28	72.3	85.56	43.06	30.68	23.98
ANCE+IRENE	39.63	48.98	61.01	77.11	88.51	48.67	34.2	26.41
MACLR	21.97	27.33	35.47	44.33	66.89	30.37	19.07	14.44
MACLR+IRENE	61.03	72.91	82.63	90.70	95.32	70.55	52.59	40.10
DPR	37.14	47.34	59.29	74.6	85.64	44.84	32.3	25.53
DPR+IRENE	52.26	64.28	75.21	83.56	88.45	61.33	45.09	35.03
TF-IDF	14.46	17.19	21.44	29.57	39.14	22.79	12.53	9.08
Zest-XML	14.56	16.39	17.48	29.28	22.85	13.97	13.29	12.67
Adam	29.31	36.08	45.04	58.39	71.76	38.3	25.19	19.13
SemSup-XC	40.56	44.39	46.81	48.95	50.09	57.45	36.11	24.67
DEXA	45.96	55.68	66.89	79.3	87.47	54.83	39.59	30.29

Model	LF-AmazonTitles-1.3M-10							
	R@3	R@5	R@10	R@30	R@100	P@1	P@3	P@5
NGAME	30.42	19.94	15.38	24.2	29.4	36.44	46.71	56.04
NGAME+IRENE	31.47	21.11	16.41	25.52	31.22	39.03	50.13	59.45
ANCE	22.38	15.14	12.02	18.14	23.32	30.72	43.66	57.76
ANCE+IRENE	22.78	15.83	12.61	19.56	24.68	32.34	45.06	58.02
MACLR	17.59	21.96	28.59	40.35	53.38	21.93	14.50	11.39
MACLR+IRENE	20.74	25.76	32.97	44.12	55.07	25.05	17.01	13.37
DPR	31.10	21.29	16.82	25.72	32.07	40.98	53.87	64.18
DPR+IRENE	31.20	21.39	16.89	25.81	32.14	40.97	53.49	63.1
TF-IDF	8.12	10.18	13.30	18.92	25.63	24.15	18.31	15.04
Zest-XML	5.42	6.35	6.87	7.89	8.67	5.58	4.71	4.22
SemSup-XC	11.28	14.73	20.04	29.26	38.27	11.68	8.41	6.85
DEXA	28.83	18.98	14.66	23.17	28.27	35.19	45.29	54.27

Model	LF-Wikipedia-500K-10							
	R@3	R@5	R@10	R@30	R@100	P@1	P@3	P@5
NGAME	53.55	58.91	65.27	74.33	82.07	46.96	22.56	15.10
NGAME+IRENE	54.45	60.19	67.03	76.12	83.51	47.01	23	15.48
ANCE	40.46	47.71	58.91	75.70	87.98	30.67	16.57	11.92
ANCE+IRENE	53.14	59.54	67.39	79.19	90.58	44.15	22.21	15.16
MACLR	51.02	58.91	68.53	81.3	91.7	39.56	21.37	15.05
MACLR+IRENE	54.34	61.94	71.01	78.69	90.79	42.91	22.70	15.78
DPR	55.28	62.70	71.20	81.22	89.43	42.90	23.02	15.96
DPR+IRENE	56.84	63.47	71.18	81.12	89.17	46.68	23.82	16.24
TF-IDF	15.43	18.64	23.89	34.66	48.24	11.53	6.29	4.60
Zest-XML	6.86	9.95	14.73	21.63	25.55	2.62	2.62	2.32
SemSup-XC	50.38	53.80	57.08	60.01	61.03	46.60	21.46	13.90
DEXA	52.45	59.15	67.37	78.04	87.36	42.76	21.91	15.07

Table 13: Generalized Zero-Shot Accuracies of different encoders when combined with IRENE. Averaged across base encoders and datasets, IRENE improves P@1, P@5, and R@10 by 14.9%, 10.4%, and 9.8%, respectively.

Model	LF-AOL-270K-10							
	R@3	R@5	R@10	R@30	R@100	P@1	P@3	P@5
NGAME	24.71	30.7	38.27	47.93	55.24	20.16	13.81	10.43
NGAME+IRENE	39.35	45.02	52.09	57.93	68.94	35.27	20.6	14.46
ANCE	31.06	39.29	49.72	58.74	74	22.63	15.86	12.25
ANCE+IRENE	37.05	43.68	51.96	59.37	73.11	30.95	19.09	13.82
MACLR	5.87	6.55	7.52	8.62	12.36	9.26	4.22	2.83
MACLR+IRENE	33.00	38.61	45.94	52.93	67.16	30.79	17.45	12.53
DPR	24.35	30.37	37.99	44.5	54.95	19.71	13.61	10.31
DPR+IRENE	39.77	45.39	52.08	57.91	68.43	35.24	20.78	14.56
TF-IDF	5.81	7.40	9.90	14.26	19.88	6.61	4.14	3.16
Zest-XML	19.67	22.18	26.57	29.8	33.39	26.34	14.71	10.19
SemSup-XC	20.98	22.55	23.92	25.21	25.86	26.12	13.97	9.08
DEXA	31.50	38.51	46.76	52.98	62.96	25.09	16.35	12.35

Model	LF-WikiHierarchy-550K-10							
	R@3	R@5	R@10	R@30	R@100	P@1	P@3	P@5
NGAME	11.24	16.70	27.08	39.85	68.14	66.19	62.64	59.25
NGAME+IRENE	14.36	21.70	36.31	54.29	81.77	87.5	83.83	80.11
ANCE	11.02	16.36	25.89	37.67	65.11	68.76	63.48	58.87
ANCE+IRENE	13.50	19.79	31.53	45	72.96	84.14	78.59	72.88
MACLR	7.23	9.89	14.31	19.63	36.31	59.44	47.20	39.93
MACLR+IRENE	14.56	22.44	38.69	59.93	88.44	88.29	87.24	84.77
DPR	11.08	16.45	26.73	39.47	68.76	65.19	61.53	58.14
DPR+IRENE	14.29	21.69	36.46	54.76	82.36	87.78	84.27	80.66
TF-IDF	6.67	8.36	10.88	15.56	22.16	64.50	42.24	32.1
Zest-XML	11.76	16.87	22.13	26.68	29.83	68.86	49.19	38.73
SemSup-XC	13.80	19.93	28.37	32.15	32.38	90.51	84.87	78.61
DEXA	13.74	21.12	36.17	53.96	81.81	76.18	76.94	75.38

Model	LF-AmazonTitles-1.3M-10							
	R@3	R@5	R@10	R@30	R@100	P@1	P@3	P@5
NGAME	17.45	22.58	30.25	42.72	54.81	45.14	39.15	34.72
NGAME+IRENE	17.82	23.10	31.04	44.12	57.03	45.81	40.04	35.61
ANCE	9.45	12.41	17.31	27.25	40.62	27.65	22.76	19.76
ANCE+IRENE	10.78	14.53	20.65	27.82	45.73	29.33	25.71	22.99
MACLR	9.13	11.76	15.99	24.57	36.57	27.50	21.86	18.60
MACLR+IRENE	9.83	13.20	18.76	25.54	43.60	26.81	22.95	20.41
DPR	14.85	19.48	26.93	35.42	55.12	38.18	32.89	29.20
DPR+IRENE	15.47	20.41	28.20	36.88	56.22	40.88	35.62	37.17
TF-IDF	13.71	16.47	20.40	27.08	34.81	16.33	9.83	7.35
Zest-XML	12.34	14.45	22.87	26.79	39.71	41.36	33.7	28.29
SemSup-XC	7.76	10.59	15.21	23.41	30.87	25.13	20.93	18.37
DEXA	18.08	23.27	30.89	38.70	54.21	48.19	40.45	35.47

Model	LF-Wikipedia-500K-10							
	R@3	R@5	R@10	R@30	R@100	P@1	P@3	P@5
NGAME	52.24	60.96	69.58	78.67	85.50	81.86	60.13	45.38
NGAME+IRENE	52.58	61.69	70.75	80.24	87.20	81.79	60.55	46.10
ANCE	29.66	35.51	43.39	56.22	71.99	42.91	27.54	20.92
ANCE+IRENE	40.49	47.77	56.66	64.44	79.01	67.68	43.17	32.14
MACLR	29.20	36.38	46.62	61.86	75.69	46.59	31.12	24.36
MACLR+IRENE	41.95	50.46	60.66	69.24	84.12	70.46	49.09	37.27
DPR	38.93	49.21	61.84	72.17	87.01	51.54	40.30	32.71
DPR+IRENE	47.56	56.70	67.19	75.62	88.51	74.98	53.81	41
TF-IDF	9.49	11.67	14.79	20.87	30.10	15.07	9.19	6.93
Zest-XML	32.24	38.01	45.15	54.31	59.32	60.16	39.33	29.21
SemSup-XC	23.44	28.83	38.08	48.01	61.30	54.20	40.72	29.58
DEXA	45.83	55.05	65.86	74.63	88.11	67.98	48.88	37.90

Table 14: Ablation study on meta-classifier generator $\mathcal{G}$ and classifier selector $\mathcal{S}$ component in IRENE on LF-WikiHierarchy-550K-10 dataset. Here IRENE is NGAME+IRENE

Method	Zero shot								Generalized							
	R@3	R@5	R@10	R@30	R@100	P@1	P@3	P@5	R@3	R@5	R@10	R@30	R@100	P@1	P@3	P@5
IRENE ($\mathcal{D}=1, K=3$)	51.97	63.46	74.21	82.52	87.55	60.93	44.77	34.67	14.36	21.70	36.31	54.29	73.80	87.50	83.83	80.11
$\mathcal{D}=2$	53.11	64.21	74.35	84.71	87.41	62.74	45.69	35.08	14.53	21.96	36.81	55.10	74.72	88.74	85.09	81.30
$\mathcal{D}=4$	51.67	62.65	73.05	79.26	87.14	61.17	44.53	34.26	14.47	21.88	36.79	55.16	74.75	88.35	84.71	80.94
$\mathcal{G}$ as Sum	37.55	47.10	59.01	73.87	84.80	45.49	32.59	25.46	11.23	16.84	27.15	39.57	56.20	68.11	64.01	60.51
$\mathcal{G}$ as wt. Sum	38.34	47.72	59.29	74.33	85.12	46.39	33.23	25.88	11.29	16.80	27.28	40.25	57.74	66.60	63.08	59.72
IRENE with Encoder Embeddings	49.10	60.60	72.36	82.31	87.66	56.79	42.14	32.92	14.25	21.49	35.89	53.75	73.33	87.17	83.12	79.29
$K=1$	52.09	63.36	73.99	80.24	87.85	60.96	44.71	34.56	14.20	21.44	35.87	53.83	81.55	86.48	82.65	78.92
$K=2$	52.59	64.05	74.75	80.76	88.13	61.15	45.05	34.88	14.31	21.62	36.19	54.22	81.91	87.11	83.44	79.73
$K=6$	48.62	59.65	71.30	78.63	84.35	57.61	41.85	32.40	14.36	21.79	36.58	54.41	73.41	87.42	84.04	80.54
$K=20$	47.63	58.45	69.95	83.44	86.60	56.93	41.07	31.78	14.42	21.88	36.73	54.6	81.36	87.86	84.55	81.00