IRENE: Improved Zero-Shot Retrieval by Extreme Meta-Classification

Sachin Yadav*
t-sacyadav@microsoft.com
Microsoft Research
India

Deepak Saini*
desaini@microsoft.com
Microsoft
USA

Anirudh Buvanesh
t-abuvanesh@microsoft.com
Microsoft Research
India

Bhawna Paliwal
bhawna@microsoft.com
Microsoft Research
India

Kunal Dahiya
kunalsdahiya@gmail.com
IIT Delhi
India

Jian Jiao
Jian.Jiao@microsoft.com
Microsoft
USA

Manik Varma
manik@microsoft.com
Microsoft Research
India

ABSTRACT

We aim to develop accurate and efficient solutions for large-scale retrieval tasks where novel (*i.e.* zero-shot) items can arrive continuously at a rapid pace. The conventional approaches embed both queries and items through a light-weight encoder and retrieve the items lying closest to the query. Such a small encoder enables efficient incorporation of new items but lacks sufficient capacity to accurately model the item semantics. To address this, we propose IRENE, an extreme meta-classifier approach which derives superior representations for novel items by enriching their text embedding with the learnt classifiers of similar, but seen items. The IRENE algorithm has minimal computational overhead, and admits efficient online deployment by encoding novel items in under 1 ms. We also propose a novel theoretical framework for analyzing the generalization performance of zero-shot retrieval, which we leverage to guide the design philosophy of IRENE. Comprehensive experiments are conducted on a wide range of XMC datasets which demonstrate that IRENE improves the zero-shot retrieval accuracy by up to 15% points in Recall@10 when combined with 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.

KEYWORDS

dense retrieval; large-scale learning; sponsored search; zero shot learning; few shot learning; extreme classification

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference'17, July 2017, Washington, DC, USA
© 2024 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Sachin Yadav, Deepak Saini, Anirudh Buvanesh, Bhawna Paliwal, Kunal Dahiya, Jian Jiao, and Manik Varma. 2024. IRENE: Improved Zero-Shot Retrieval by Extreme Meta-Classification. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Sponsored search is a match-making system between users and advertisers that aims to optimize user experience when searching for knowledge while helping advertisers reach interested users. Users encode their intent in short text queries. Similarly, advertisers bid on short-text keywords relevant to their ads. A critical component in any sponsored search stack is matching user queries to these advertiser keywords. Most search engines currently follow different semantics to do this matching called match-types. 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. Further, since this matching task sits towards the head of the ads retrieval pipeline, gains here propagate downstream.

A major challenge in this is retrieving novel keywords, which have limited or no historical click data associated with them. Existing dense retriever methods like ANCE rely solely on the keyword's textual content to generate representations. However, for short novel keywords, the text alone may be insufficient. On the other hand, extreme classification methods like NGAME learn keyword classifiers that encode useful collaborative signals beyond just text similarity. But these require some training data, which is lacking for novel keywords.

This paper proposes IRENE to address this challenge of retrieving novel keywords. The key insight is that classifiers of seen (non-novel) keywords capture collaborative signals that can be transferred to novel keywords. IRENE represents a novel keyword by enriching its textual embedding with classifiers of similar seen keywords. This allows incorporating collaborative knowledge from

seen keywords to compensate for the lack of training data on novel keywords.

Motivational experiments illustrate that text embeddings and classifier representations encode diverse signals. Query predictions using seen keywords' classifiers were found to be 59% different on average from predictions using just text embeddings. This validates that solely text-based similarity is insufficient and motivates combining text and classifier information in a principled manner.

IRENE has three main components - a text encoder, seen keyword classifiers, and a combiner module. The text encoder maps keywords to embedding space. The seen keyword classifiers are learned to encode collaborative signals. Finally, the combiner non-linearly merges a novel keyword's text embedding with classifiers of its top nearest seen neighbors. It employs self-attention with custom type embeddings to effectively integrate these diverse representations. IRENE is trained in two stages for computational and generalization efficiency. First, the text encoder is trained independently using contrastive loss. Then keeping the encoder fixed, the seen keyword classifiers and combiner module are jointly trained. This modular approach allows the combiner to harmonize the heterogeneous knowledge in text embeddings and classifiers.

Experiments were conducted on proprietary and public datasets spanning sponsored search, product recommendations, and Wikipedia document tagging. On all applications, IRENE outperformed state-of-the-art baseline methods including dense retrievers like ANCE and extreme classifiers like NGAME. Notably, IRENE also achieved significant gains over methods specifically designed for novel keywords, such as ZestXML, NCM, and ADAM. On a commercial search engine, it increased click-through rate by 4.2% and decreased quick-back rate by 0.9%, retrieving more relevant keywords. On public benchmark applications, IRENE was found to be 1.5-3% more accurate than state-of-the-art Dense Retrieval algorithms. Additionally, the dual stage training employed by IRENE allows it to be trained by utilizing any embedding-based dense retriever. IRENE can hence make use of advancements in teacher training, negative-mining, loss functions etc. that researchers in the dense retrieval community seek to improve

The paper makes the following contributions

(1) We identify that classifier representations encode diverse signals compared to text embeddings.
(2) We propose combining text embeddings and classifiers of similar seen keywords to represent novel keywords. The combiner is designed modularly to work with different encoders and allows joint training to improve novel keyword retrieval.
(3) IRENE can incrementally improve predictions for one-shot keywords given newly revealed data, without fine-tuning the full model. This reduces latency and complexity.
(4) IRENE's efficient two-stage training enables scaling to 200M+ queries, allowing validation via A/B tests on live search traffic.

2 RELATED WORKS

The popular approaches for keyword retrieval in Sponsored Search applications include DR, XC, and generative algorithms.

Dense Retrieval: Classical information retrieval (IR) methods including BM25 [29] make use of lexical features to return the subset of most relevant items. On the other hand, dense retrieval methods (DR) aim to learn a feature encoder that embeds relevant items close to queries as compared to irrelevant ones. DR approaches including DPR [18] and ANCE [36] have been demonstrated to comprehensively outperform BM25 on several retrieval applications including passage retrieval [7]. Recently proposed algorithms further improved over ANCE by eliminating false positives [28], introducing token-level interactions [21] and incorporating graph neural networks [39]. While more accurate, RocketQA [28], SimCSE [12], ColBERT [21], and GraphFormers [39] can also be more computationally demanding as compared to ANCE. Interestingly, DR methods can incorporate zero-shot items out-of-the-box, however, the item representations are constrained by the text which may be insufficient, especially in the case of short-text items [11]. XC algorithms present a way to learn improved item representations via classifiers as described below.

Extreme Classification: Classical extreme classifications algorithms focus on learning scalable and accurate classifiers for a large number of seen items [10, 11, 15, 16, 20, 24, 27, 31, 40]. XC algorithms have found applications in multiple applications including document tagging [3], product-to-product recommendation [17, 19], and sponsored search [8, 9]. Unfortunately, classical XC methods are designed to handle items that are tagged with at least a few training queries and hence miss out on novel items which constitute a large chunk of items in sponsored search applications.

However, XC methods such as Semsup-XC [1], MACLR [37], and ZestXML [14] have been proposed recently to incorporate zero-shot and few-shot items. In particular, ZestXML [14] employs a sparse transformation to embed relevant items close to the query as compared to irrelevant ones. However, ZestXML's accuracy can be lacking especially on novel labels [1]. SemSup-XC handles novel items through a hybrid matching module that matches input instances to class descriptions using a combination of semantic and lexical similarity. However, Semsup-XC employs a shortlist based on sparse bag-of-words features that can grow to be large in order to retrieve hard negatives. In addition, Semsup-XC relies on web-scraped meta-data to boost performance on unseen items. While it may be feasible to procure such meta-data at smaller scales, it often becomes a challenging task when the number of items is in the scale of millions. Furthermore, MACLR incorporates self-supervised learning tasks of Inverse Cloze Task and SimCSE for encoder pre-training. Finally, IRENE can be 2-12% more accurate than zero-shot XC methods on publicly available benchmark datasets.

Zero/few-shot learning: A ton of methods have been specifically proposed to incorporate novel labels, *i.e.,* labels with at most one train point. Label attributes or prototypes can be used to transfer the knowledge from seen items to novel items [30, 34, 38]. For example, ESZSL [30] assumes that attributes like "striped", "four-legged", *etc.,* that can describe both seen and unseen animal classes are provided for a task to predict animals present in an image. Hence such methods may not apply to large-scale retrieval settings like advertiser bid keyword retrieval as collecting or generating attributes for millions of items can be a daunting task in itself. NCM [26] and ADAM [41] employ centroids to compute representations of novel items with at least one training query. These methods can scale to a large number of novel items, however, their performance may be lacking on seen labels as demonstrated in the experiments

Table 1: Average number of different query prediction for a given keyword using its text-based and classifier representation, both obtained from NGAME algorithm. Values for different number of top predictions are shown

Top-k Considered	% of different predictions
3	70.66
5	65.2
10	62.13
30	60.86
50	59.72
80	59.25
100	58.98

section. The representation for novel items can also be learned via classifier synthesis where the classifiers of novel items are synthesized using classifiers of seen items [22, 25] or phantom items [5]. These methods attempt to synthesize classifiers over fixed features from a pre-trained model. For instance, COSTA [25] assume that item ground-truth co-occurrence of novel item to seen items can be obtained by either a web search engine or a user-provided input. However, scraping such data for a large number of novel items is infeasible in practice. Changpinyo16 learn the phantom items with the objective of aligning the classifier and text-embedding vector spaces. However, IRENE takes a different approach and tries to harmonize the diverse information present in the two vector spaces and obtain a combined representation for novel item. Moreover, these methods employ a linear combination while synthesizing classifiers. While these relaxations may help the methods in terms of scalability but miss out on potential accuracy gains provided by feature fine-tuning and a more expressive combination. In particular, IRENE employs feature fine-tuning and a novel attention-style combiner which yielded 2% more accurate item representations as compared to simpler combinations as proposed by LAF [22].

Finally, class incremental learning approaches (CIL) fine-tune the encoder based on the training data points of novel items, wherever available. However, fine-tuning to few shot labels may lead to catastrophic forgetting issues [41]. We, therefore, compare IRENE against a theoretical baseline where we pool the revealed data for novel items with the training data and retrain a DR model from scratch. Moreover, exemplar-based [2, 6] or exemplar-free [35, 42] algorithms attempt to mitigate the catastrophic forgetting, however, these may get expensive when dealing with a large number of novel items. Further, frequently fine-tuning an existing model in production can add additional latency and logistical complexities. This is further compounded by the necessity to rebuild the ANNS data structure from scratch. IRENE achieves state-of-the-art accuracy while being computationally efficient as a novel item can be encoded within 1 ms plus IRENE can utilize incremental ANNS data structure thus making it suitable for sponsored search application with millions of novel keywords.

3 MOTIVATION

In this section, we provide some experiments to motivate the IRENE algorithm. The Key idea in DR is to learn two embedding functions $\mathcal{E}_q$ and $\mathcal{E}_i$ for mapping a query, x and an item, y to some D-dimensional vector space. Specifically, that $\mathcal{E}_q(x) \in V$ and $\mathcal{E}_i(y) \in V$, where $V = \mathbb{R}^D$. At the same time, the core problem in extreme multi-label classification is to learn a function $f(x) \rightarrow \mathcal{Y}$ that maps the input representation $\mathcal{E}_{xc}$ of x to the most relevant subset of labels from the label set $\mathcal{Y}$. State-of-the-art deep extreme classification algorithms solve this problem by learning a language encoder to map queries to some say d-dimensional vector space, i.e. $\mathcal{E}_{xc}(x) \in \mathbb{R}^D$. The algorithms would further train d-dimensional classifiers, $w_l \in \mathbb{R}^D$ one per item. Given sufficient data for an item, the learned classifier, which is not solely dependent on the item text alone is able to encode collaborative signals. At prediction time, the items whose classifiers give the input query the largest score are predicted. This operation can be represented as $\max_{j=1}^{|\mathcal{Y}|} w_j^T \mathcal{E}_{xc}(x)$. This max-k operation can be speeded up using ANNS data structures built using the appropriate distance metric. But please note that this is essentially the pathway adopted by DR algorithms at prediction time. Hence, extreme classifiers can be thought of as dense retrievers with classifiers as item representations.

With this context in mind, we conduct an experiment to analyze the text-based embedding for an item and the classifier trained for the items in an extreme classification algorithm. We train an NGAME encoder on the KeywordPredicton-5M dataset with 5M keywords as a dense retriever. We further train NGAME classifiers on the same dataset. Please note that the NGAME algorithm freezes the encoder while training the classifiers. Therefore, for a given query at prediction time, its representation is the same whether we were to use the text-based embedding of a keyword output by the NGAME encoder or its classifier representation. As an experimental query set, we sample random 10M queries from a set of top 600M queries asked on the search engine after deduping with the queries contained in the KeywordPrediction-5M training dataset. We use the NGAME encoder to get their embeddings and build a DiskANN algorithm on them. Then, as a thought experiment, we try to solve the problem of, given a keyword from the 5M set, which top 100 queries out of these 10M, when asked on the search engine, would lead to click on it. We obtain the two sets of predictions by inferencing the text-based NGAME encoder representation of the keyword and the NGAME classifier representation using the same query ANNS index. Please note that all the pieces, except the item representation, are the same between the two set of predictions. Table 1 shows the fraction of different predictions in top top-k predictions from the two sets as k is varied. In particular, on average around 59% of the queries retrieved by using a classifier representation of a keyword are different from the queries retrieved using its text-based representation. This experiment shows that the text-based and classifier representations of an item can encode diverse information and their combination might potentially lead to better representation for an item.

However, it is not possible to train classifiers for zero-shot items. For one-shot items with just one supervision query revealed, it is not possible to train their classifiers accurately. IRENE tries to bridge this gap by utilizing the classifiers trained for similar items for such novel items instead. Please note that this definition can be easily extended to rare items having < k training queries, where k can vary depending on the retrieval application. However, in this

Table 2: Performance when the associated items were (in a hypothetical scenario) revealed from the ground truth. Validation novel items were sampled from the N-Amazon-1.3M dataset to simulate these experiments. The accuracy degraded when the ground truth associations were progressively replaced by neighbors computed on the basis of similarity between the novel and seen item's encoder representations.

Configuration	P@1	P@3	P@5
Ground truth associations	33.80	22.66	17.62
Replace 1 neighbor	33.04	22.22	17.30
Replace 2 neighbors	32.30	21.73	16.93
Replace 3 neighbors	31.57	21.23	16.47
NGAME (no combiner)	30.42	19.94	15.38

work, we focus only on zero-shot and one-shot scenarios. The next section describes the algorithmic details in IRENE.

4 IRENE

Algorithm 1 Getting representation for novel item in IRENE. **Input**: Novel item u, **Output**: Enriched representation $\mathcal{E}^+(u)$

$\mathcal{E}_\theta(u) \leftarrow$ Encoder(u) ▷ Get embedding
$S^u_\theta \leftarrow$ ANNS$(\mathcal{E}_\theta(u), \mathcal{E}_\theta(l))$ ▷ Get shortlist
if zero-shot **then**
  $S^u_w \leftarrow$ Refine$(S^u_\theta, \mathcal{E}_\theta(u), w_l, \tau_0)$
else if one-shot **then**
  $S^u_w \leftarrow$ Refine$(S^u_\theta, \mathcal{E}_\theta(u), w_l, \tau_0)$
  $S^u_{w'} \leftarrow$ Refine$(S^u_\theta, \mathcal{E}_\theta(x_u), \tau_1)$
  $S^u_w \leftarrow$ MaxVote$(S^u_w, S^u_{w'})$
end if
$\{l_1, l_2, l_3\} \leftarrow$ TopK$(S^w_u, 3)$
$\{w_{l_1}, w_{l_2}, w_{l_3}\} \leftarrow$ GetClassifiers$(\{l_1, l_2, l_3\})$
$\mathcal{E}^+(u) \leftarrow$ Combiner$(\mathcal{E}_\theta(u), w_{l_1}, w_{l_2}, w_{l_3})$
return $\mathcal{E}^+(u)$

Notations: Let $\mathcal{Y}_s$ denote the set of seen items encountered during the training phase. Note that $L = |\mathcal{Y}_s|$ remains constant. Correspondingly, let $\mathcal{Y}_u$ be the set of novel items which are made available at inference time, *i.e.,* $|\mathcal{Y}_u|$ can keep on changing over time.

Architecture: The IRENE architecture is comprised of three components, the feature encoder $\mathcal{E}_{\boldsymbol{\theta}}$, 1-vs-all classifiers for each seen label $\mathbf{W} = \{\mathbf{w}_l\}_{i=1}^{|\mathcal{Y}_s|}$, and the combiner $\mathcal{C}_{\boldsymbol{\phi}}$. Please refer to Figure 1 which illustrates IRENE's components while representing a novel item. The encoder embeds a query $\mathbf{x}$ (or item) in a D-dimensional space, *i.e.,* $\mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x}) \in \mathbb{R}^D$. IRENE is a generic algorithm suitable for several real-world applications including product-to-product recommendations, document tagging, and matching user queries to advertiser keywords. Thus, the feature encoder may be chosen depending on the application - the default encoder is set to a 6-layer DistilBERT [32] architecture.

4.1 Item representation

Popular DR methods including ANCE use the textual description $\mathbf{z}_l$ of an item l using the feature encoder, *i.e.,* $\mathcal{E}_{\boldsymbol{\theta}}(\mathbf{z}_l)$. While novel items may be naturally incorporated in DR models, the representation may be lacking especially for short-text items comprised of just a few tokens such as keywords in Sponsored Search application. On the other hand, XC algorithms make use of a classifier-based representation, *i.e.,* $\mathbf{w}_l$, which allows the item representation to move beyond text-based representation. However, it is challenging to learn classifiers for a novel item with no or limited data. Thus, IRENE proposes a combiner $\mathcal{C}_{\boldsymbol{\phi}}$ that represents the item l, whether seen or unseen, using $\mathcal{E}_{\boldsymbol{\theta}}(\mathbf{z}_l)$ and a sub-set of classifiers $\{\mathbf{w}_j\}_{j \in \mathcal{S}^l}$. Here, $\mathcal{S}^l$ is a set of similar seen items for item l discussed in detail in the next sections. The combiner allows IRENE to generalize better to novel items. In particular, IRENE yielded 3% accurate predictions over NGAME, ANCE, and SimCSE for predicting novel items, demonstrating the effectiveness of the proposed combiner described below.

The Combiner: The combiner, $\mathcal{C}_{\boldsymbol{\phi}}$ employs a single-layer self-attention block with novel type-embeddings to represent an item. In particular, the representation for an item is obtained as $\mathcal{E}^+(l) = \text{MLP}(\text{Self-Attention}(\mathcal{E}_{\boldsymbol{\theta}}(\mathbf{z}_l) + \mathbf{t}_{enc}, \mathbf{w}^l_1 + \mathbf{t}_{clf}, \mathbf{w}^l_2 + \mathbf{t}_{clf}, .., \mathbf{w}^l_{|\mathcal{S}^l|} + \mathbf{t}_{clf}))$. Here, t_{enc} and t_{clf} are type-embeddings, one each for classifier and encoder, and are learned like positional embeddings as in BERT. MLP is a linear transformation. Also, $\mathcal{E}_{\boldsymbol{\theta}}(\mathbf{z}_l)$ is the encoder representation computed on the basis of the text, and $\{\mathbf{w}^l_j\}_{j \in \mathcal{S}^l}$ represent the classifiers of associated seen items for item l. The associated seen items S^l may be computed in a variety of ways depending on the class of items, *viz.,* seen, zero-shot, and one-shot as described in the next sub-section. Please see Fig. 1 illustrating the combiner. IRENE's representation of a novel item can be computed within 1 ms making it suitable for sponsored search application.

Associations for seen items: The associations for seen items are available from ground truth, *i.e.,* y_{lm} is observed when both $l, m \in \mathcal{Y}_s$. However, employing associations only on the basis of ground truth leads to misalignment between seen and novel items as such associations are not available for novel items. Moreover, ground truth may bring in some undesirable or noisy associations. For instance, in Wikipedia dataset, categories of "Academics of King's College London" and "1878 deaths" have the article of "Johann Joseph Hoffmann" in common. However, associating "Academics of King's College London" with "1878 deaths" may end up providing spurious signals and may not help in improving the item representation. Thus, IRENE refines the associations based on the agreement of the ground truth with encoder-based representations, for the two items l and m. In practice, at max top three associated items were selected from the ground-truth co-occurring items based on text similarity, *i.e.,* $|\mathcal{S}^l| <= 3$. Increasing $|\mathcal{S}^l|$ did not offer any substantial gains in our experiments.

Associated seen items for novel items: Let u denote a novel item with $\mathbf{z}_u$ as a textual description. Moreover, a novel item may be (optionally) tagged with a data point with textual description $\mathbf{x}_u$ in some scenarios. Please note that these correspond to the

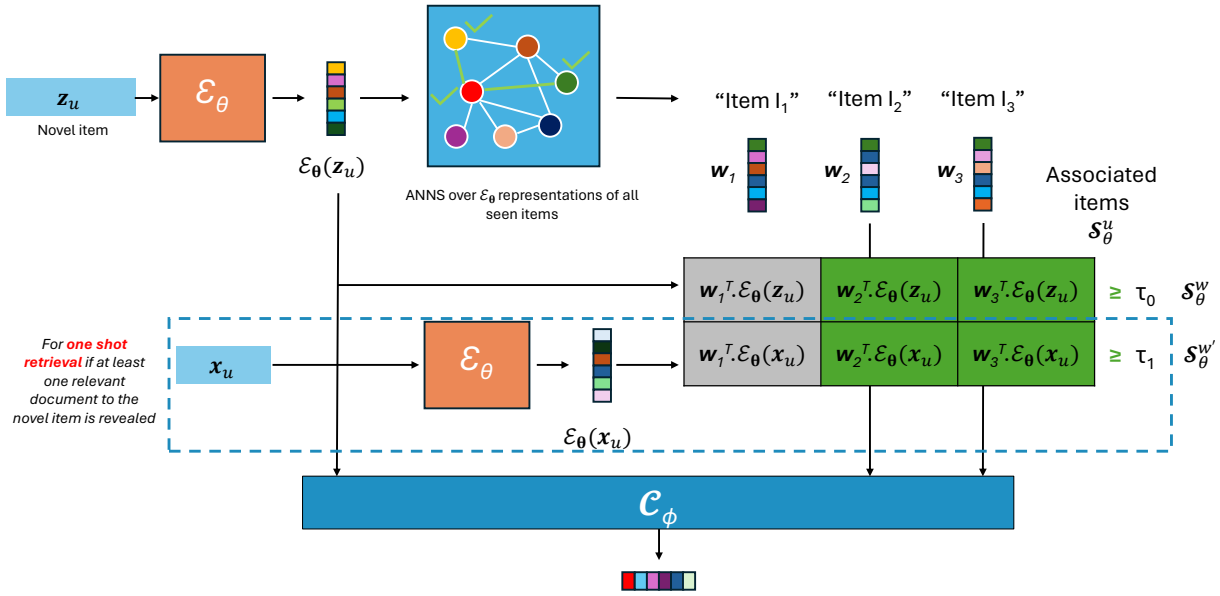

Figure 1: The representation for a novel item u with item text z_u and (optional) revealed document x_u is computed using a self-attention based combiner function C_ϕ. C_ϕ expects u's encoder based embedding ($\mathcal{E}_{\boldsymbol{\theta}}(z_u)$) alongside the classifier representations of associated seen items ($\{w_1, w_2, w_3\}$) as input. The associated seen items are computed on the basis of $\mathcal{E}_{\boldsymbol{\theta}}(z_u)$ for zero-shot novel items and $\{\mathcal{E}_{\boldsymbol{\theta}}(z_u), \mathcal{E}_{\boldsymbol{\theta}}(x_u)\}$ for novel one-shot items. Please refer to text for more details. Best viewed in color.

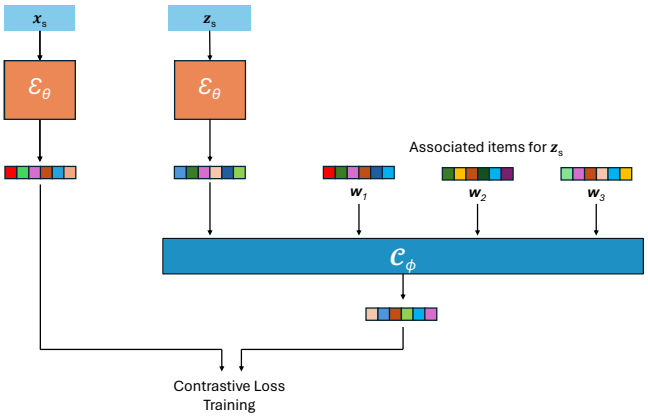

Figure 2: IRENE's training pipeline. The point is embedded using the encoder $\mathcal{E}_{\boldsymbol{\theta}}$ and the item is embedded using the combiner $C_{\boldsymbol{\phi}}$ and passed to a contrastive loss. Please refer to the text for more details. Best viewed in color.

zero-shot and one-shot settings, respectively. A straightforward way to compute the associations with seen items is to make use of encoder-based representation. The top-k associated seen items can be computed via maximum inner product search (MIPS) as $\mathcal{S}^u_{\boldsymbol{\theta}} = \arg\max_k \{\mathcal{E}_{\boldsymbol{\theta}}(z_l)^\top \mathcal{E}_{\boldsymbol{\theta}}(z_u); l \in \mathcal{Y}_s\}$ for a novel item u. Top K items are retrieved so that $|\mathcal{S}^u_{\boldsymbol{\theta}}| = K$. The shortlist, $S^u_{\boldsymbol{\theta}}$, obtained in the previous step may contain items that are textually similar based on $\mathcal{E}_{\boldsymbol{\theta}}$ but are not necessarily relevant to u. To address this, a refinement step is introduced with the aim of pruning away dissimilar items. Hence, $\mathcal{S}^u_{\boldsymbol{\theta}}$ was further refined by pruning away irrelevant items by deploying the classifier representation of seen items to get $\mathcal{S}^u_w$. Specifically, we compute $\mathcal{S}^u_w = \{l | \mathbf{w}_l^\top \mathcal{E}_{\boldsymbol{\theta}}(z_u) \geq \tau_0; l \in \mathcal{S}^u_{\boldsymbol{\theta}}\}$. This refinement process serves the purpose of the zero-shot setting. Additionally, in the case of a one-shot setting, where a query associated with the novel item is also available, we get another shortlist by evaluating item classifiers on the revealed query. Specifically, another shortlist can be obtained by performing the refinement step $\mathcal{S}^u_{w'} = \{l | \mathbf{w}_l^\top \mathcal{E}_{\boldsymbol{\theta}}(x_u) \geq \tau_1; l \in \mathcal{S}^u_w\}$. Note this is the only step in the algorithm which uses the revealed query associated with novel item. We further do a max-voting between the two shortlists hence obtained by refining based on the label text embedding, S^w_u and based on the revealed query, $S^{w'}_u$. K, τ_0, and τ_1 are hyper-parameters of the algorithm. It should be noted that the $|\mathcal{S}^u| \in [0, 3]$ can vary with the item u. This procedure is explained in pseudo-code 1.

Interestingly, in some cases, $\mathcal{S}^u$ may be empty and the combiner reduces to a linear function applied over the item's encoder-based representation $\mathcal{E}_{\boldsymbol{\theta}}(z_u)$. The intuition here is to only pass the classifiers of seen items that are deemed relevant to the novel items based on the chosen criterion. A simulation indicated that the performance may further improve if better-associated items are provided. In particular, a small mutually exclusive validation set of novel items were sampled from the train set whose associations are available from ground truth (say $\mathcal{S}^*$). $\mathcal{S}$ was initialized with the associations based on item embeddings, *i.e.*, $\hat{\mathcal{S}}$. The performance improved when neighbors in $\mathcal{S}$ were gradually replaced with ground truth-based associations in $\mathcal{S}^*$ (please refer to Table 2).

4.2 Training

IRENE aims to learn its model parameters $\{\boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{W}\}$ by optimizing

$$\min_{\boldsymbol{\theta},\mathbf{W},\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta},\mathbf{W},\boldsymbol{\phi}) = \sum_{i=1}^{N} \sum_{\substack{p\in\mathcal{Y}_s \\ y_{ip}=+1}} \sum_{n\in\mathcal{Y}_s} \ell(\mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x}_i), C_{\boldsymbol{\phi}}(\mathcal{E}_{\boldsymbol{\theta}}(\mathbf{z}_p), \mathbf{U}^p), C_{\boldsymbol{\phi}}(\mathcal{E}_{\boldsymbol{\theta}}(\mathbf{z}_n), \mathbf{U}^n)) \quad (1)$$

where, i refers to a query, p refers to a positive item ($y_{ip} = +1$), n refers to a irrelevant item ($y_{in} = -1$). Moreover, $\mathbf{U}^p = \{\mathbf{w}_j \mid j \in \mathcal{S}^p_{\boldsymbol{\theta}}\}$ is the set of classifier belonging to associated items for positive item p. Recall that $\mathcal{S}^p_{\boldsymbol{\theta}}$ denotes the set of associated items for item p. Similarly, $\mathbf{U}^n$ is the set of classifier belonging to associated items for item n. ℓ can be a loss function such as Triplet loss. Unfortunately, optimizing (1) induces a cost $O(NDL)$ which can be prohibitive for a large number of items. Moreover, the associated items a seen item is computed on the basis of the item's encoder representation which will keep on changing as ℓ is optimized. Thus IRENE employs a modular pipeline where the encoder is trained in the first phase and then classifiers + combiner are learned in the second stage with a fixed encoder.

Stage I (Learning the encoder): Stage I treats the item embeddings as classifiers and learn $\mathcal{E}_{\boldsymbol{\theta}}$ in a Siamese fashion. This allows IRENE to forgo explicit dependency on L and hence it can be efficiently trained with a proper negative mining strategy. The negative mining algorithms aim to cut down the training cost by restricting the item set for each query to $O(\log L)$. The negative mining approach based on batching similar train queries together in a mini-batch proposed by NGAME was employed for negative sampling as it has been demonstrated to yield state-of-the-art results while being computationally lightweight. It should be noted that other negative mining approaches including ANCE can be readily incorporated in IRENE. In particular, $\boldsymbol{\theta}$ is learned by optimizing

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \sum_{\substack{p\in\mathcal{Y}_s \\ y_{ip}=+1}} \sum_{n\in\mathcal{N}_i} [\mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x}_i)^\top \mathcal{E}_{\boldsymbol{\theta}}(\mathbf{z}_n) - \mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x}_i)^\top \mathcal{E}_{\boldsymbol{\theta}}(\mathbf{z}_p) + \gamma]_+ \quad (2)$$

where, γ is margin in triplet loss, p is a positive label and $\mathcal{N}_i$ is the shortlist of negative labels for query i. IRENE employs the Adam optimizer in both stages to perform gradient descent. In practice, training of Stage I completes within a few hours for all publicly available benchmark datasets.

Stage II (Learning the classifier and combiner): The Stage II freezes the encoder ($\boldsymbol{\theta}$) and jointly learns the classifier ($\mathbf{W}$) and combiner ($\boldsymbol{\phi}$). Freezing the encoder allows IRENE to compute the associated items $\mathcal{S}^l$ once and then freeze them thereafter thereby improving the scalability. In principle, fine-tuning the encoder during Stage II may yield marginal gains, however, at the cost of significantly increased training time. Specifically, model parameters $\mathbf{W}, \boldsymbol{\phi}$ are learnt by optimizing

$$\min_{\mathbf{W},\boldsymbol{\phi}} \mathcal{L}(\mathbf{W},\boldsymbol{\phi}) = \sum_{i=1}^{N} \sum_{\substack{p\in\mathcal{Y}_s \\ y_{ip}=+1}} \sum_{n\in\mathcal{N}_i} [\mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x}_i)^\top \mathbf{w}_n - \mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x}_i)^\top \mathbf{w}_p + \gamma_1]_+ + [\mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x}_i)^\top C_{\boldsymbol{\phi}}(\mathcal{E}_{\boldsymbol{\theta}}(\mathbf{z}_n), \mathbf{U}^n) - \mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x}_i)^\top C_{\boldsymbol{\phi}}(\mathcal{E}_{\boldsymbol{\theta}}(\mathbf{z}_p), \mathbf{U}^p) + \gamma_2]_+ \quad (3)$$

where γ_1 & γ_1 are margin terms in triplet loss and $\hat{\mathbf{x}}_i$, $\hat{\mathbf{z}}_p$, $\hat{\mathbf{z}}_n$ are fixed encoder based representations for i^{th} data point, positive item p and negative item n. Note that such a formulation allows IRENE to generalize better to novel items without sacrificing on seen items. As a result, IRENE could be 2% more accurate in R@100 as compared to state-of-the-art DR models in a generalized setting. Please refer to Section 5 for more details.

4.3 Inference

The embedding of a document t at inference time is computed as $\mathbf{e}_t = \mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x}_t)$ where x_t is the document text. An ANN$\mathcal{S}$ index A is built over the item representations from our model, *i.e.*, A is built over $\{\mathcal{E}^+(l); l \in \mathcal{Y}_s \cup \mathcal{Y}_u\}$. To get the most relevant items for t, A is queried with $\mathbf{e}_t$. The computational complexity of IRENE's inference is $\Omega(B + D\log(|\mathcal{Y}_s| + |\mathcal{Y}_u|))$, where B is the cost of encoder and D is representation dimensionality. Note that the IRENE continues to integrate novel items by computing their representations and adding them to ANN$\mathcal{S}$ index [33] as and when novel items appear in the system, thereby offering a promising solution for applications like sponsored search where novel items are frequently encountered and re-training the model can be expensive with large number of items.

5 EXPERIMENTS

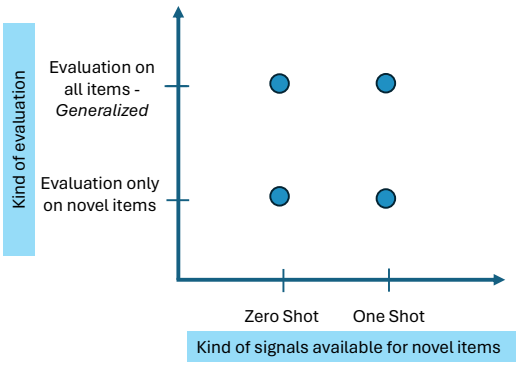

Figure 3: Four settings explored in this work based on kind of evaluation and kind of signals available for novel items

In this section, we describe **(i)** datasets on which we benchmark IRENE, **(ii)** Baselines against which IRENE is compared, and **(iii)** Evaluation settings and metrics.

Datasets: IRENE was designed with the advertiser keyword application in mind. Therefore, we conduct both offline and online experiments for this application. Furthermore, we validate IRENE's effectiveness in acquiring enhanced representations for new items

Table 3: Results on KeywordPrediction-5M dataset. Semsup-XC-NGAME is a theoretical baseline where NGAME encoder from the table is fine-tuned on revealed query associations for one-shot items according to Semsup-XC

Method	R@30	R@50	R@100	P@30	P@50	P@100
Evaluation only on novel items						
IRENE-ZeroShot	**34.79**	**49.81**	**77.32**	**78.70**	**71.76**	**60.18**
NGAME encoder	33.76	48.07	74.13	76.52	69.32	57.67
SimCSE	31.37	42.87	72.45	76.3	68.40	56.32
IRENE-OneShot	**35.87**	**50.95**	**78.46**	**79.75**	**72.84**	**61.20**
NCM	32.94	46.74	71.72	74.79	67.34	55.48
Semsup-XC-NGAME	34.43	49.06	74.99	77.53	70.15	58.04
Generalized evaluation						
IRENE-ZeroShot	**37.22**	**54.00**	**85.86**	**80.21**	**73.72**	**62.72**
NGAME encoder	36.10	52.04	82.02	78.01	71.89	60.01
SimCSE	38.37	54.87	88.45	76.36	68.40	57.32
IRENE-OneShot	**37.71**	**54.55**	**86.64**	**80.68**	**74.11**	**62.84**
NCM	34.98	50.12	78.17	75.69	69.51	55.70
Semsup-XC-NGAME	36.13	51.73	80.17	77.69	70.51	58.70

Table 4: 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
Evaluation only on novel items						
IRENE- ZeroShot	**30.53**	**42.00**	**66.68**	**45.64**	**43.40**	**40.11**
M1	28.79	39.64	62.19	44.24	41.96	38.72
M2	19.11	25.84	39.33	31.48	28.93	25.69
M3	19.93	26.49	38.99	35.72	33.11	29.63
M4	24.74	32.82	48.46	39.10	36.60	33.25

Table 5: Expert judges labeling results on KeywordPrediction-5M dataset

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

Table 6: Latencies required to perform different steps to get IRENE representation for a novel item y_u. Steps 1, and 4 are done on a single V100 GPU, steps 2 and 3 on 96 CPU cores. All numbers are reported for batched operations.

Step	Time (in μs)
Calculate the embedding $\mathcal{E}_{\boldsymbol{\theta}}$	24
Calculate encoder emb shortlist $S_u^{\boldsymbol{\theta}}$	600
Refine the shortlist using classifers to $S_u^{\boldsymbol{\theta}}$	0.46
Apply the Combiner function $C_{\boldsymbol{\phi}}$	0.3

by evaluating it on public datasets. These tasks encompass similar product recommendations on Amazon and category annotation for Wikipedia articles.

Table 7: Results on public datasets with different algorithms.

Method	N-Amazon-1.3M			N-Wikipedia-500K		
	P@1	P@5	R@5	P@1	P@5	R@5
	Evaluation only on novel items					
IRENE-ZeroShot	**31.47**	**16.41**	**31.22**	**47.01**	**15.48**	**60.19**
ANCE	22.38	12.02	23.32	30.67	11.92	47.71
NGAME Encoder	30.42	15.38	29.40	46.96	15.10	58.91
SemSup-XC	11.68	6.85	14.73	46.60	13.90	53.80
MACLR	21.93	11.39	21.96	39.56	15.05	58.91
ZestXML	5.58	4.22	6.35	2.62	2.62	2.32
TF-IDF	24.15	15.04	10.18	11.53	6.29	4.60
IRENE-OneShot	**31.92**	**16.67**	**31.68**	**47.90**	**15.79**	**61.42**
NGAME-OneShot (re-trained)	30.55	15.39	29.43	47.37	15.33	59.86
Semsup-XC-OneShot	13.96	6.98	15.49	46.25	13.77	53.67
NCM OneShot	26.49	48.57	28.94	40.43	13.60	53.72
ADAM OneShot	20.80	11.71	25.73	39.79	13.39	52.97
	Generalized evaluation					
IRENE	**45.81**	**35.61**	**23.10**	81.79	**46.10**	**61.69**
ANCE	27.65	19.76	12.41	42.91	20.91	35.51
NGAME Encoder	45.14	34.72	22.58	**81.86**	45.38	60.96
Semsup-XC	25.13	18.37	10.59	63.92	26.59	37.22
MACLR	27.50	18.60	11.76	46.59	24.38	36.38
ZestXML	22.87	26.79	39.71	45.15	54.31	59.32
NCM	41.36	28.29	14.45	60.16	29.21	38.01
TF-IDF	16.33	7.35	16.47	15.07	6.93	11.67
IRENE-OneShot	**45.83**	**35.66**	**23.12**	81.80	**46.16**	**61.77**
NGAME OneShot (re-trained)	45.31	34.87	22.66	**82.22**	45.60	61.17
Semsup-XC OneShot	22.48	15.66	10.03	60.97	25.40	36.16
NCM OneShot	28.41	19.19	13.66	61.89	30.04	46.27
ADAM OneShot	19.13	15.26	11.50	59.54	27.77	43.46

Benchmark datasets: We create zero and one-shot datasets for evaluation on Amazon product recommendations and category annotation for Wikipedia pages by using the datasets LF-AmazonTitles-1.3M and LF-Wikipedia-500K from extreme classification repository [4]. Please refer to the supplementary material for details on dataset creation and data statistics.

Evaluation settings and metrics: Fig. 3 describes the four evaluation setting considered: evaluation on only novel items for zero and one-shot items; and generalized evaluation for zero and one-shot items. For a detailed description of the 4 settings please refer to the supplementary. Performance is evaluated on standard metrics including Precision@k (P@k), and Recall@k (R@k) at different truncation levels k. The truncation level is decided by the application. For the Keyword prediction application, an algorithm's performance depends on the fraction of good keywords in the top-k keywords retrieved where large values of k are considered depending on the deployment infrastructure. Hence P@k is reported for $k \in \{30, 50, 100\}$.

Baselines: We compare IRENE with three classes of methods **(i)** Dense Retrieval algorithms like NGAME and ANCE. **(ii)** Extreme classification algorithms like MACLR, Semsup-XC, and ZestXML that are specifically designed to handle zero and few-shot scenarios and **(iii)** Continual learning baselines like NCM and Adam. The key idea in continual learning is to utilize new data while minimizing catastrophic forgetting. In addition, we also consider a theoretical baseline where the new data is pooled with the training data, and an NGAME dense retrieval encoder is trained on the pooled data end-to-end. Please note that generally, it is not possible to re-train a model frequently(say nightly) as new signals come in. For dense retrieval, algorithms are trained to utilize a DistilBERT model as the language encoder. The hyper-parameters of all baseline algorithms

are set as suggested by their authors wherever applicable and by fine-grained validation otherwise.

Hyper-parameters: IRENE's tunable hyper-parameters over and above the hyper-parameters used by any DR algorithm, include the shortlist size K and the filtration thresholds for refined shortlists τs. As discussed in the supplementary material, high-quality connections to seen items is essential for IRENE to obtain their high-quality representations. Hence, these metrics were set by k-fold cross validation. All other hyper-parameters were set to default values across the datasets. Refer to supplementary material for the hyper-parameter values for IRENE on different datasets.

Advertiser bid keyword retrieval: We discuss both offline and online results for the application.

Online Results IRENE was flighted on a leading search engine to perform A/B tests on live search engine traffic. However, we also do an explicit comparison of IRENE against leading proprietary and public DR algorithms in production. More specifically, we sample 100M advertiser keywords that had come into the system after the time period for which the training data scraping is done. We further take some of the top dense retrieval encoders deployed in production and compare IRENE against them for recommending keywords from this 100M set for a sampled set of queries. The algorithms' names for this comparison are anonymized for IP reasons. The results are shown in table 4. IRENE was found to be at least 4% better than the next best dense retriever in R@100. As novel items stream into the system, it might be necessary to frequently encode the items and include them in the ANNS index. Table 6 shows that IRENE adds only minimal overhead on top of a language encoder and can get the item representation in less than one ms. Algorithms that fine-tune the trained model on revealed data such as Semsup-XC need to rebuild the ANNS index from scratch. IRENE can instead make use of updatable ANNS [33] algorithms similar to any DR algorithm. Further, as users interact with the search engine as the live A/B test is conducted, IRENE is compared against the large control ensemble of diverse algorithms. This ensemble not only contains DR algorithms but also leading generative, graph-based, and IR algorithms. Performance is measured in terms of live performance metrics. 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 didn't find the ad relevant and left the ad landing page quickly) by 4.2% and 0.9% respectively. This indicates value creation for users who were shown more relevant ads by IRENE. Additionally, IRENE could increase the keyword density (average number of keywords that survive the quality control and relevance filters) by 7.8%, which validates the quality of predictions made. IRENE could also achieve a click efficiency of 150% meaning that for every 2% increase in ad impressions, ads selected by IRENE got 3% more clicks. Further, IRENE could match queries like "grainger" and "bitwarden" to advertiser keywords like "industrial supply" and "password manager" respectively. Please note that these predictions, not based on text matching, were not made by any in-production algorithm.

Offline Results To conduct experiments offline experiments, KeywordPrediction-5M dataset was created by mining the logs of a search engine for a specific time period. User-typed queries and the bid keyword corresponding to surfaced advertisements yielded query-keyword training pairs. The dataset, named KeywordPrediction-5M, had around 5M items and 220M training queries Please note that we finetune the trained NGAME encoder using the methodology prescribed in the Semsup-XC work [1] to obtain the Semsup-XC-NGAME baseline. However, please note that it is not desirable to fine-tune models deployed in production as it incurs latency and complexity costs. As shown in table 3, IRENE was found to be at least 3% better in R@100 and at least 2% better in P@30 for evaluation only on zero-shot items. Similarly, when evaluation is done only on one-shot items, IRENE was found to be around 3% and 2% better in R@100 and P@30 respectively. Similar trends were observed for the generalized evaluation setting.

Table 8: Ablation study on Combiner Functions and Contribution of Clasifier Representations to IRENE on N-Amazon-1.3M dataset

Method	R@5	R@10	R@30	R@5	R@10	R@30
	Zero shot			One shot		
Combiner - Sum	30.36	38.13	49.29	30.74	38.71	50.36
Combiner - Weighted Sum	29.85	37.12	47.58	30.01	37.32	47.93
Combiner - Max	29.60	36.83	47.04	29.95	37.39	48.12
IRENE	31.22	39.03	50.13	31.68	39.72	51.14
IRENE with Encoder Embeddings	30.64	38.21	49.18	31.03	38.76	49.95

Similar product recommendation on Amazon Experiments are conducted on the N-AmazonTitles-1.3M dataset for this application. IRENE was found to be at least 2% better than the baseline algorithms in R@5 when the evaluation is done only on novel items. Similarly, IRENE outperformed algorithms that could make use of the revealed data by at least 2% in R@5. Similar trends are seen in generalized evaluation for both zero-shot and one-shot scenarios.

Wikipedia categories prediction Experiments are also conducted on the N-Wikipedia-500K dataset for predicting categories for a Wikipedia page. On this application, IRENE again outperforms the baseline algorithms. For zero-shot retrieval of novel articles, IRENE achieves over 6% higher recall than extreme classifiers like SemSup-XC and ZestXML. The gains are maintained in the one-shot setting where IRENE incorporates the revealed training article per novel category during inference. For a detailed evaluation of other metrics please refer to Table 11 (in the supplementary)

The consistent improvements demonstrate IRENE's ability to generalize across different use cases by learning improved representations for novel items

6 ABLATIONS

Experiments were conducted to understand the contributions of different components of the proposed method in IRENE.

Utilizing Encoder vs Classifier Representations: As discussed in sections 1 and 3, classifier representations of associated seen items in addition to encoder representation of a novel item contain useful diverse signals for encoding the novel item. Experiments were done to identify the importance of the classifier representations. Specifically, encoder representations instead of learnt classifiers were utilized for obtaining representations of associated seen items while learning the combining function C_{ϕ}. As shown in Table 8, learning classifier representations as proposed in IRENE leads to 2% gain in R@30 over IRENE trained with encoder embeddings only on N-Amazon-1.3M dataset.

Combining Functions For obtaining a robust representation of a novel item, it is important to carefully *combine* the signals - encoder representation of the novel item and collaborative information from the classifier representations of associated seen items. Keeping the inputs fixed, the combiner C_ϕ was changed to simpler alternatives for vector pooling suggested in previous works such as sum, max, and weighted-sum poolings. In weighted sum pooling, the weights for the different positions are learned.

As can be observed in Table 8, the self-attention with type embeddings combiner proposed in IRENE performs 2% more accurately than the best-performing sum-based combination.

Modularization with Different Encoder We evaluate the modularization of proposed algorithm IRENE with respect to different encoders by comparing the performance of IRENE when added to ANCE based encoder on N-Amazon-1.3M dataset. The addition of IRENE to existing ANCE encoder leads to an improvement in performance in all Recall based metrics with 2% improvement in R@30. Refer to the supplementary material for more details.

7 CONCLUSIONS AND FUTURE DIRECTION

In this paper, we present IRENE architecture to address one of the key challenges in Sponsored Search, which is the retrieval of novel keywords. IRENE involves minimal computational onverhead over language-encoders making it an ideal solution for such large scale settings s. Extensive online and offline experiments demonstrate the superior effectiveness of IRENE compared to current baseline methods.

REFERENCES

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

[2] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio. 2019. Gradient based sample selection for online continual learning. In *NeurIPS*.

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

[4] K. Bhatia, K. Dahiya, H. Jain, P. Kar, A. Mittal, Y. Prabhu, and M. Varma. 2016. The Extreme Classification Repository: Multi-label Datasets & Code. http://manikvarma.org/downloads/XC/XMLRepository.html

[5] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. 2016. Synthesized Classifiers for Zero-Shot Learning. In *CVPR*.

[6] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. S. Torr. 2018. Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence. In *ECCV*.

[7] N. Craswell, B. Mitra, E. Yilmaz, D. Campos, and E. M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. arXiv:2003.07820

[8] 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*.

[9] 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*.

[10] 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*.

[11] 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*.

[12] T. Gao, X. Yao, and D. Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *EMNLP*.

[13] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgeting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211* (2013).

[14] 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*.

[15] 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*.

[16] 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*.

[17] 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*.

[18] 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*.

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

[20] 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*.

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

[22] 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.

[23] Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *In Psychology of learning and motivation* 24 (1989), 109–165.

[24] 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*.

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

[26] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. 2013. Distance-Based Image Classification: Generalizing to New Classes at Near-Zero Cost. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 11 (2013), 2624–2637. https://doi.org/10.1109/TPAMI.2013.83

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

[28] 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.

[29] S. Robertson and H. Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3 (2009), 333–389.

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

[31] 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*.

[32] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv* (2019).

[33] 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]

[34] J. Snell, K. Swersky, and R. S Zemel. 2020. Prototypical Networks for Few-shot Learning. In *NeurIPS*.

[35] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, Perot V., J. Dy, and T. Pfister. 2022. Learning to Prompt for Continual Learning. arXiv:2112.08654 [cs.LG]

[36] 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*.

[37] 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]

[38] W. Xu, Y. Xian, J. Wang, B. Schiele, and Z. Akata. 2020. In *NeurIPS*.

[39] J. Yang, Z. Liu, S. Xiao, C. Li, D. Lian, S. Agrawal, A. Singh, G. Sun, and X. Xie. 2021. GraphFormers: GNN-nested Transformers for Representation Learning on Textual Graph. In *NeurIPS*.

[40] 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*.

[41] D.-W. Zhou, H.-J. Ye, D.-C. Zhan, and Z. Liu. 2023. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need.

[42] F. Zhu, X.-Y. Zhang, C. Wang, F. Yin, and C.-L. Liu. 2021. Prototype Augmentation and Self-Supervision for Incremental Learning. In *CVPR*.

Table 9: Dataset Statistics.

Dataset	# Train Qs	# novel Test Qs	# Gen Test Qs	# Train Items	# novel Items
KeywordPrediction-5M*	220,845,427	5,022,052	5,022,052	4,999,996	5,435,443
N-AmazonTitles-1.3M	2,225,354	624,830	970,237	1,174,739	130,526
N-Wikipedia-500K	1,781,890	271,620	783,743	450,963	50,107

Table 10: Hyper-parameters for different datasets for IRENE.

Dataset	Encoder($\mathcal{E}_{\theta}$) lr	Encoder($\mathcal{E}_{\phi}$) lr	Classifiers(w) lr	# epochs for $\mathcal{E}_{\theta}$	# epochs for $\mathcal{E}_{\phi}$+w	Margin γ_1, γ_2	K	τ_0
KeywordPrediction-5M	0.0001	0.0002	0.001	50	70	0.1,0.3	100	0.75
N-AmazonTitles-1.3M	0.0002	0.0002	0.001	300	120	0.1,0.3	200	0.15
N-Wikipedia-500K	0.0002	0.0002	0.001	40	120	0.1,0.3	200	0.2

Table 11: Results on benchmark datasets with different algorithms.

Method	N-Amazon-1.3M								N-Wikipedia-500K							
	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
	Evaluation only on novel items															
IRENE-ZeroShot	**25.52**	**31.22**	**39.03**	**50.13**	**59.45**	**31.47**	**21.11**	**16.41**	**54.45**	**60.19**	67.03	76.12	83.51	**47.01**	**23.00**	**15.48**
ANCE	18.14	23.32	30.72	43.66	57.76	22.38	15.14	12.02	40.46	47.71	58.91	75.70	87.98	30.67	16.57	11.92
NGAME Encoder	24.20	29.40	36.44	46.71	56.04	30.42	19.94	15.38	53.55	58.91	65.27	74.33	82.07	46.96	22.56	15.10
SemSup-XC	11.28	14.73	20.04	29.26	38.27	11.68	8.41	6.85	50.38	53.80	57.08	60.01	61.03	46.60	21.46	13.90
MACLR	17.59	21.96	28.59	40.35	53.38	21.93	14.50	11.39	51.02	58.91	**68.53**	**81.30**	**91.70**	39.56	21.37	15.05
ZestXML	5.42	6.35	6.87	7.89	8.67	5.58	4.71	4.22	6.86	9.95	14.73	21.63	25.55	2.62	2.62	2.32
TF-IDF	8.12	10.18	13.30	18.92	25.63	24.15	18.31	15.04	15.43	18.64	23.89	34.66	48.24	11.53	6.29	4.60
IRENE-OneShot	**25.86**	**31.68**	**39.72**	**51.14**	**60.62**	**31.92**	**21.41**	**16.67**	**55.38**	**61.42**	**68.69**	**78.08**	**85.50**	**47.90**	**23.39**	**15.79**
NGAME-OneShot (re-trained)	24.27	29.43	36.48	46.76	56.03	30.55	20.00	15.39	54.42	59.86	66.37	75.28	82.85	47.37	22.92	15.33
Semsup-XC-OneShot	12.55	15.49	19.76	26.93	35.27	13.96	9.07	6.98	50.21	53.67	57.04	60.04	61.02	46.25	21.24	13.77
NCM-OneShot	23.49	28.94	36.70	48.57	60.32	26.49	17.52	13.62	48.03	53.72	60.94	71.26	80.22	40.43	19.97	13.60
Adam-OneShot	20.61	25.73	33.08	44.49	55.85	20.70	14.80	11.71	47.22	52.97	60.37	70.65	79.97	39.79	19.60	13.39
	Generalized evaluation															
IRENE	**17.82**	**23.10**	**31.04**	**44.12**	**57.03**	**45.81**	**40.04**	**35.61**	**52.58**	**61.69**	**70.75**	**80.24**	**87.20**	81.79	**60.55**	**46.10**
ANCE	9.45	12.41	17.31	27.25	40.62	27.65	22.76	19.76	29.66	35.51	43.39	56.22	71.99	42.91	27.54	20.92
NGAME Encoder	17.45	22.58	30.25	42.72	54.81	45.14	39.15	34.72	52.24	60.96	69.58	78.67	85.50	**81.86**	60.13	45.38
Semsup-XC	7.76	10.59	15.21	23.41	30.87	25.13	20.93	18.37	34.22	37.22	39.37	40.72	41.10	63.92	38.84	26.59
MACLR	9.13	11.76	15.99	24.57	36.57	27.50	21.86	18.60	29.20	36.38	46.62	61.86	75.69	46.59	31.12	24.38
ZestXML	12.34	14.45	22.87	26.79	39.71	41.36	33.7	28.29	32.24	38.01	45.15	54.31	59.32	60.16	39.33	29.21
NCM	5.53	13.82	18.69	27.98	40.10	29.80	23.24	19.86	38.88	45.69	54.20	66.13	76.54	60.51	39.50	29.58
TF-IDF	13.71	16.47	20.40	27.08	34.81	16.33	9.83	7.35	9.49	11.67	14.79	20.87	30.10	15.07	9.19	6.93
IRENE-OneShot	**17.83**	**23.12**	**31.06**	**44.17**	**57.14**	**45.83**	**40.07**	**35.66**	**52.64**	**61.77**	**70.84**	**80.37**	**87.37**	81.80	**60.61**	**46.16**
NGAME-OneShot (re-trained)	17.52	22.66	30.33	42.78	54.88	45.31	39.33	34.87	52.44	61.17	69.82	78.96	85.81	**82.22**	60.46	45.60
Semsup-XC-OneShot	7.64	10.03	13.87	20.97	28.76	22.48	18.25	15.66	32.89	36.16	38.73	40.52	41.08	60.97	36.70	25.40
NCM-OneShot	10.58	13.66	18.55	28.00	40.48	28.41	22.36	19.19	39.47	46.27	54.78	66.61	76.84	61.89	40.20	30.04
Adam-OneShot	8.52	11.50	16.06	24.72	36.47	19.13	17.30	15.26	37.09	43.46	51.61	63.00	73.81	59.54	37.35	27.77

Evaluation metrics Evaluation was performed on Precision@k (P@k), Recall@k (R@k). For a query i, the predicted score vector $\hat{\mathbf{y}}_i \in R^L$ and ground truth vector $\mathbf{y}_i \in \{0,1\}^L$:

$$P@k = \frac{1}{k} \sum_{l \in rank_k(\hat{\mathbf{y}})_d} y_{dl}$$

$$R@k = \frac{1}{||\mathbf{y_i}||_0} \sum_{l \in rank_k(\hat{\mathbf{y}})_d} y_{dl}$$

Datasets creation details: The results are presented on N-LF-Amazon-1.3M and N-Wikipedia-500K. These datasets were derived from LF-AmazonTitles-1.3M and LF-Wikipedia-500K available on the extreme classification repository [4]. Zero-shot and one-shot train-test splits were created in the following manner:

Zero-Shot split: 10% of items were randomly selected to form the novel item set. The remaining 90% of items with their associated queries formed the training corpus. The novel test set was created based on connections between test queries and novel items, while the generalized test set matched the extreme classification repository's version.

One-shot split: Here each novel item is provided one query, which is randomly chosen from the connections between novel item set and train queries. Note that IRENE's was found to be quite robust to the sampled query for novel items. The novel and generalized test sets are the same in both zero and one-shot settings.

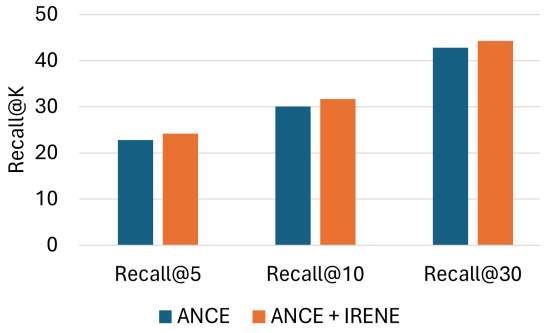

Figure 4: Plot shows the improvement in Recall for novel items over ANCE encoder with the addition of IRENE on N-Amazon-1.3M

Evaluation Settings The problem setting considered in this paper can be divided on the basis of two axes. Specifically, the two axes considered are:

What kind of signals are made available for novel items:

- Zero-Shot setting: This setting focuses on the case when novel items $\mathcal{Y}_u$ have no click signal made available. This setting assumes that the novel item which has been included for prediction has only its text available.
- One-Shot setting: Within this setting, we concentrate on items for which we possess exactly one associated clicked query. Unlike the zero-shot setting, where only the textual content of items is utilized for representation generation, in this setting, we assume that exactly one ground-truth query for the novel item is also revealed.

What kind of evaluation is being done:

- Evaluation only on novel items: this setting assumes that prediction has to be done only on novel items, i.e, Y_s
- Generalized evaluation: in this evaluation setting, the prediction set is composed of both items seen during training and novel items $Y_s \cup Y_u$. In this work, we explicitly consider the evaluation only on novel items because novel items are the focus of this work and sometimes, depending on the distribution of the generalized item set at prediction time, the algorithm can show accuracy gains by focusing solely on the seen items. At the same time, the phenomena of catastrophic forgetting leading to better accuracy on novel items but still, pooer accuracy on seen items is well studied [13, 23]. To guard against such a possibility, both of these 2 types of evaluations are considered.

A cartesian product of the 2 settings on the 2 axes leads us to a total of 4 settings considered in the paper.