Large Language Model as a Teacher for Zero-shot Tagging at Extreme Scales

Jinbin Zhang[1], Nasib Ullah[1], Rohit Babbar[1,2],
[1]Aalto University [2]University of Bath
{jinbin.zhang, nasibullah.nasibullah, rohit.babbar}@aalto.fi, rb2608@bath.ac.uk

Abstract

Extreme Multi-label Text Classification (XMC) entails selecting the most relevant labels for an instance from a vast label set. Extreme Zero-shot XMC (EZ-XMC) extends this challenge by operating without annotated data, relying only on raw text instances and a predefined label set, making it particularly critical for addressing cold-start problems in large-scale recommendation and categorization systems. State-of-the-art methods, such as MACLR (Xiong et al., 2022) and RTS (Zhang et al., 2022), leverage lightweight bi-encoders but rely on suboptimal pseudo labels for training, such as document titles (MACLR) or document segments (RTS), which may not align well with the intended tagging or categorization tasks. On the other hand, LLM-based approaches, like ICXML (Zhu and Zamani, 2024), achieve better label-instance alignment but are computationally expensive and impractical for real-world EZ-XMC applications due to their heavy inference costs. In this paper, we introduce LMTX[1] (**L**arge language **M**odel as **T**eacher for e**X**treme classification), a novel framework that bridges the gap between these two approaches. LMTX utilizes an LLM to identify high-quality pseudo labels during training, while employing a lightweight bi-encoder for efficient inference. This design eliminates the need for LLMs at inference time, offering the benefits of improved label alignment without sacrificing computational efficiency. Our approach achieves superior performance and efficiency over both LLM and non-LLM based approaches, establishing a new state-of-the-art in EZ-XMC.

1 Introduction

Extreme Multi-label Text Classification (XMC) is the task of assigning relevant labels to documents from an extensive label space, often comprising hundreds of thousands to millions of possible labels (Bhatia et al., 2016). XMC is widely applied in real-world scenarios such as product-to-product recommendations, product search (Chang et al., 2021a), labeling Wikipedia pages (Babbar and Schölkopf, 2017), and categorizing Amazon products (Jiang et al., 2021). Despite its widespread use, existing supervised XMC methods depend heavily on expert-annotated labels or user-annotated labels, with the label set fixed during both training and inference. Furthermore, supervised XMC faces two challenges. First, obtaining annotations is difficult due to the sheer scale of the label space, which makes it challenging for annotators to select relevant labels, often resulting in incomplete or missing labels (Qaraei et al., 2021; Schultheis and Babbar, 2021; Schultheis et al., 2022; Wydmuch et al., 2021; Jain et al., 2016; Schultheis et al., 2024). Second, the dynamic emergence of new labels, especially in cold-start scenarios adds further complexity. Conventional XMC methods are poorly equipped to handle unseen labels during inference, limiting their capacity to adapt to the evolving and dynamic nature of the label space.

There are two distinct settings for zero-shot extreme classification: (i) Generalized Zero-Shot Extreme Multi-label Learning (GZXML) (Gupta et al., 2021), which enables models to predict unseen labels but still relies on annotated training data, making it unsuitable for scenarios lacking labeled data, such as cold-start problems; and (ii) Extreme Zero-Shot Multi-label Text Classification (EZ-XMC) (Xiong et al., 2022; Zhang et al., 2022), which handles unseen labels without requiring any annotated data. In this work, we adopt the EZ-XMC setting to address cases where labeled data is unavailable, new labels emerge dynamically, and mainly focus on tagging application tasks.

Current EZ-XMC methods predominantly focus on training robust bi-encoders by leveraging pseudo-positive labels generated from the documents themselves. This approach enables the en-

[1]The Github link: https://github.com/xmc-aalto/LMTX

Proceedings of the 31st International Conference on Computational Linguistics, pages 3465–3478
January 19–24, 2025.

coding of label texts into embeddings via a sentence encoder, facilitating efficient retrieval aligned with document embeddings. Crucially, this methodology eliminates the need for training datasets to encompass the entire label spectrum. For instance, MACLR (Xiong et al., 2022) constructs instance-pseudo label pairs using (content, title) combinations from documents, while RTS (Zhang et al., 2022) randomly splits documents and selects two spans to form such pairs (Figure 1). However, these methods often overlook the direct semantic alignment between the document and pseudo-label pairs. For instance, a segment generated by the RTS might not be relevant to another segment if they are located too far apart within the same document. Moreover, the pseudo-labels may not adequately reflect the domain of the predefined label set, leading to a mismatch between the target task and the generated training pairs.

Large Language Models (LLMs) have recently exhibited remarkable reasoning and zero-shot capabilities across diverse NLP tasks (Bonifacio et al., 2022; Saad-Falcon et al., 2023; Ma et al., 2023; Qin et al., 2023; Sun et al., 2023; Dai et al., 2023; Hou et al., 2023; Sachan et al., 2023, 2022). Nevertheless, only a few notable exceptions (Zhu and Zamani, 2024; Xu et al., 2023b; Liu et al., 2024) have been explored in the context of XMC problems. This limited adoption is primarily due to the substantial computational overhead associated with deploying LLMs, especially given the large search space typical of XMC tasks. Additionally, the inference phase for XMC problems can become prohibitively expensive when using heavy LLM models. To address this limitation, we propose a novel relevance assessment strategy that leverages an LLM to judiciously select high-quality pseudo labels from a curated label set for each document. This approach enables the training of a lightweight bi-encoder model that inherits the LLM's knowledge while avoiding the inference-time computational burden. Our contributions can be summarized as follows:

- LMTX introduces a novel training approach for bi-encoders, emphasizing a curriculum-based method that dynamically adjusts based on the relevance feedback from an LLM by leveraging its zero-shot learning abilities.
- The proposed LMTX requires less training data because there is a higher correlation between the pseudo-labels and documents, resulting in higher-quality training pairs. Consequently, our approach achieves better performance while maintaining similar or reduced training time compared to traditional methods for some large datasets.
- The proposed LMTX enables the lightweight deployment by using only the bi-encoder to generate embeddings for documents and labels during the prediction. LLM models are not involved in the prediction process. LMTX significantly outperforms current state-of-the-art methods for the tagging task, demonstrating comprehensive advancements in performance metrics.

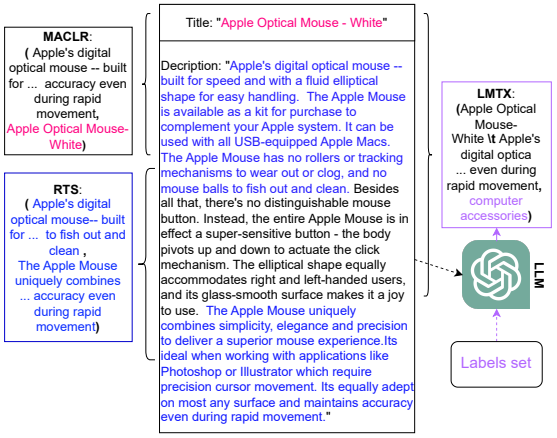

Figure 1: An example of how to construct training pairs using state-of-the-art methods MACLR (Xiong et al., 2022) and RTS (Zhang et al., 2022). MACLR utilizes the 'Title' of a document to generate pseudo labels, while the 'Description' serves as the training document. Conversely, RTS forms its training pairs by selecting two random segments from the 'Description'. Differently, our proposed model, LMTX, adopts a more refined approach. It selects 'computer accessories' as a pseudo positive label from a predefined set, a choice validated by the LLM model.

2 Background

Problem Definition: Let's denote $X_i \in \mathcal{X}$ as the text for an instance in a particular domain; i.e., X_i could be the textual description for a product on Amazon. Unlike the supervised XMC, the key characteristic of the EZ-XMC setting is that we do not have the corresponding well-annotated labels Y_i for each training instance X_i. However, besides having the original text of instances $\{X_i\}_{i=1}^{N}$, we also have access to the predetermined labels along with their texts, i.e., we have $\{l_k\}_{k=1}^{L}$. We refer to this

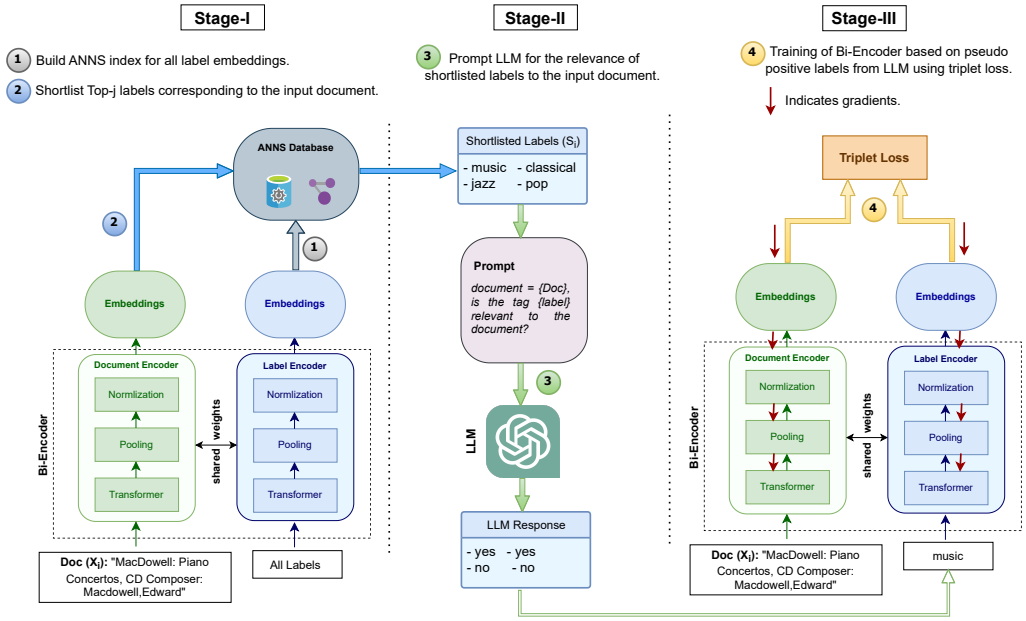

Figure 2: The process of getting feedback from LLM model for training the bi-encoder. First, for a given document, the (pre) trained bi-encoder and ANNS are employed to create a short-list of potential labels. Next, the LLM assesses the relevance between the labels in this shortlist and the document. Finally, the selected labels are utilized to further train the bi-encoder.

collection of predetermined labels as the "labels set". The goal of EZ-XMC, which is the one that we consider in this paper, is to assign the document $X_i \in \mathcal{X}$ to set of labels $\{l_j\} \subseteq \{l_k\}_{k=1}^{L}$ that are relevant to the document. To achieve this objective, the task requires learning a mapping function from text to embeddings for both $\{X_i\}_{i=1}^{N}$ and $\{l_k\}_{k=1}^{L}$, so that the $\{l_k\}_{k=1}^{L}$ can be retrieved in the same space as $\{X_i\}_{i=1}^{N}$ by comparing their embedding similarity. The mapping function is denoted as $\mathcal{E}_\theta : \mathcal{X} \to \mathbb{S}^{D-1}$, where θ represents the training parameters, $\mathcal{E}$ represents the encoder for documents and labels, and $\mathbb{S}^{D-1}$ is the D-dimensional unit sphere. The mapping function is typically implemented as a bi-encoder, where both the text of documents and labels are embedded within $\mathbb{S}^{D-1}$.

Bi-Encoder Model: We employ a bi-encoder architecture, $\mathcal{E}_\theta$, to generate embeddings for both document and label text. The model consists of two encoders with shared weights: one for documents and another for labels. The document and label embeddings are represented as $\mathcal{E}_\theta(X_i)$ and $\mathcal{E}_\theta(l_k)$, respectively, where X_i is the document and l_k is the label text. The relevance score between document X_i and label l_k is computed via cosine similarity between their embeddings. The bi-encoder we use is based on the Distill-BERT transformer (Sanh et al., 2019) and depicted in Figure 2.

3 Training the Bi-encoder from the Feedback of LLM

Training Process Overview: Our methodology adopts an iterative framework, encompassing three distinct stages within each cycle. Initially, we embed all documents and labels, subsequently constructing an Approximate Nearest Neighbor Search (ANNS (Malkov and Yashunin, 2018)) over the label embeddings to retrieve a refined set of label candidates for each document. In the second stage, the LLM is deployed to scrutinize these candidates, effectively identifying pseudo positive labels. The final stage involves training the bi-encoder model using the labels identified in the preceding stage. Figure 2 illustrates the mechanism through which the bi-encoder incorporates feedback from the LLM and progresses through training regimen.

Data Embedding & Shortlist Generation (stage-I): The LLM model demonstrates zero-shot ability in determining relevance between two text segments (Ma et al., 2023). However, this approach encounters challenges when applied to a vast array of labels, as in our context. Specifically, the computational complexity involved in assessing the relevance between each document and every label in a large set becomes formidable, being $\mathcal{O}(NL)$ in complexity. This can be quite prohibitive, even for a dataset with a moderate number ($\mathcal{O}(10^3)$) of

instances and labels. To mitigate this, our strategy involves condensing the label space presented to the LLM. We utilize (pre) trained bi-encoder to process the document and label text into embeddings and utilize ANNS to efficiently select the top-j most relevant labels for each document. These selected labels, denoted as $S_i = \{l_{i1}, l_{i2}, ..., l_{ij}\}$, constitute a focused subset for subsequent processing.

LLM Model as a Teacher (stage-II): Once we obtain the label shortlist S_i for the i-th document, we can employ the LLM as a teacher to determine the relevance between the document and the top-j labels in a shortlist. Let X_i denote a particular document and l_{ik} be its k-th label in the shortlist. To assess the relevance between X_i and l_{ik}, we instruct the LLM with the question, "document = $\{X_i\}$, is the tag $\{l_{ik}\}$ relevant to the document? answer yes or no". If the LLM outputs "Yes", we consider l_{ik} to be relevant to X_i. Conversely, if the model outputs "No", we consider l_{ik} as an unrelated label and discard it. We keep all the labels from the shortlist that received a positive feedback ("yes") from the LLM. Then, we use these selected relevant labels to train the bi-encoder model. A detailed discussion of different prompts used for the LLM can also be found in Appendix A.6.

Algorithm 1 Training the bi-encoder with the feedback from LLM teacher (LMTX)

Input: Initial bi-encoder $\mathcal{E}_\theta$, LLM model $\mathcal{M}_{LLM}$, data instances $\{X_i\}_{i=1}^N$, labels set $\{l_k\}_{i=1}^L$, dev set instances $\{X_j\}$, and number of cycles T
Output: Trained bi-encoder $\mathcal{E}_\theta$

$c = 0$.
while $c < T$ **do**
  Compute $\mathcal{E}_\theta(X_i), \mathcal{E}_\theta(l_k)$ for all $\{X_i\}_{i=1}^N$ and $\{l_k\}_{i=1}^L$
  Retrieve top labels $S_i = ANNS(\mathcal{E}_\theta(X_i), \mathcal{E}_\theta(l_k)_{k=1}^L)$ for each X_i
  Fetch pseudo positive labels $P_i^+ = \mathcal{M}_{LLM}(X_i, S_i)$ for all X_i
  for i=0 to N_batches **do**
    Sample a mini_batch $B_i = \{X_i, P_i^+\}$ where, $|B_i| = m$
    Update $\mathcal{E}_\theta$ using mini-batch B_i, loss $\mathcal{L}$ and AdamW optimizer.
  end for
  Evaluate $\mathcal{E}_\theta$ with $\mathcal{M}_{LLM}$ on the dev set $\{X_j\}$ and obtain $P@1$ over pseudo labels.
  if $P@1$ does not improve on dev dataset **then**
    Stop training cycle
  end if
  $c = c + 1$
end while
return model $\mathcal{E}_\theta$

Training Bi-Encoder with Pseudo Positive Labels (stage-III): To train the bi-encoder, we follow the training procedure in (Dahiya et al., 2023). Out of the labels identified by the LLM as the pseudo positives, we choose only one of the pseudo positive labels for each document during the training process. This is shown to help in achieving faster convergence in the earlier work (Dahiya et al., 2023). Regarding the negatives, which we need to compute the instance-wise loss, we use in-batch negative sampling, in which the negatives for a document come from the pseudo positive labels of other documents in the same batch. Our analysis in Section 5 shows that using labels which are rejected by the LLM, as hard negatives, leads to degradation in prediction performance.

For the label l_k, the predicted relevance score between document X_i and l_k is computed through the cosine similarity $\langle \mathcal{E}_\theta(X_i), \mathcal{E}_\theta(l_k) \rangle$, and triplet loss is used to train the bi-encoder (Schroff et al., 2015a; Manmatha et al., 2017; Dahiya et al., 2023):

$$\mathcal{L} = \sum_{i=1}^{N} \sum_{k'} [\langle \mathcal{E}_\theta(X_i), \mathcal{E}_\theta(l_{k'}) \rangle - \langle \mathcal{E}_\theta(X_i), \mathcal{E}_\theta(l_p) \rangle + \gamma]_+ \tag{1}$$

where γ is the margin, the k' stands for the index of hard negative labels from the mini batch, $l_{k'}$ and l_p correspond to the text of the negative labels and the pseudo positive label.

As training progresses, the bi-encoder gradually improves, leading to an enhancement in the quality of labels within the shortlist and increased relevance to the corresponding document. During training, we evaluate the model on the development dataset and choose the best one based on performance evaluated by the LLM since under the EZ-XMC setting one does not have access to annotated ground-truth labels. If there is no performance improvement on the development set, training is halted, so the number of cycles is actually determined by the performance on the development dataset. The pseudo code of the proposed algorithm LMTX, for training the bi-encoder model with feedback from LLM, is presented in Algorithm 1.

Inference: The model's inference procedure is analogous to the formation of the shortlist during training, as depicted in Stage-I of Figure 2. We build the MIPS (Johnson et al., 2019) over these label embeddings, which implements the efficient maximum inner product search. For each document, we employ its embedding as a query to retrieve the top-m labels, which ultimately serve as the predicted results. The use of MIPS[2] in the infer-

[2]https://github.com/facebookresearch/faiss

ence process ensures a sublinear time complexity for each instance. The label embedding extraction and construction of MIPS index are performed just once, hence amortizing the cost of this step.

Dataset	N	N_{test}	N_{label}	L_N
EURLex-4K	15,511	3,803	3,956	20.79
Wiki10-31K	14,146	6,616	30,938	8.52
AmazonCat-13K	1,186,239	306,782	13,330	448.57
LF-WikiSeeAlso-320K	693,082	177,515	312,330	4.67
LF-Wikipedia-500K	1,813,391	783,743	501,070	24.75

Table 1: Statistical overview of the datasets. N: total number of training samples, N_{test}: number of test samples, N_{label}: total number of unique labels, L_N: average number of samples per label.

4 Experiments

Datasets and Evaluation Metrics: We utilized five tagging datasets for evaluation: EURLex-4k, Wiki10-31k, and AmazonCat-13K were obtained from the XLNet-APLC repository[3], while the remaining datasets were downloaded from the extreme classification repository[4]. Table 1 provides comprehensive statistical information for all datasets. To optimize computational resources, we constrained the training data for AmazonCat-13K, LF-WikiSeeAlso-320K, and LF-Wikipedia-500K to 30,000 documents each. In contrast, baseline models utilize the entire dataset.

We employ the commonly used evaluation metrics (Reddi et al., 2019; Chang et al., 2021b; Zhang et al., 2022) for the EZ-XMC setting: $Precision@k$ and $Recall@m$. Further details on the evaluation metrics and implementation can be found in the Appendix A.2 and A.1 respectively.

Baselines: We have incorporated state-of-the-art EZ-XMC models as our baselines. The baseline contains unsupervised pseudo-labels methods: MACLR (Xiong et al., 2022) and RTS (Zhang et al., 2022). Unsupervised pre-trained embeddings and encoders: GloVe (Pennington et al., 2014), Inverse Cloze Task (ICT) (Lee et al., 2019) and MPNet (Song et al., 2020). Sentence matching: Sent-BERT (Reimers and Gurevych, 2019) and SimCSE (Gao et al., 2021). Pre-trained retrieval bi-encoder: Msmarco-distilbert (Reimers and Gurevych, 2021). LLM-based methods: ICXML (Zhu and Zamani, 2024). To assess the baseline performance of LF-WikiSeeAlso-320K and LF-Wikipedia-500k, we obtained the results from (Zhang et al., 2022). As for the other baselines, we acquired their performance by executing the respective baseline.

Comparison with standard baselines: In Table 2, we present a comparative analysis of our model's performance against other models. Notably, our LMTX model demonstrates substantial improvements in both $Precision@m$ & $Recall@m$, especially for datasets like EURLex-4k, Wiki10-31k, AmazonCat-13k, and LF-Wikipedia-500k. Particularly striking are the results in LF-Wikipedia-500k and AmazonCat-13K, where our model shows an increase of 31% and 37%, respectively, for $P@1$. In addition, our results on LF-WikiseeAlso-320k are competitive with those of the leading models, despite the unique nature of this task, which focuses on identifying related Wikipedia titles rather than traditional tagging. Moreover, Table 6 presents a comparison of the training time and computational resources required for LMTX relative to other methods, further underscoring the efficiency of our approach. These results strongly indicate that our approach is both computationally efficient and highly effective in zero-shot scenarios, capable of addressing diverse tagging and categorization tasks with state-of-the-art performance.

Comparison with LLM-based baseline: We compared our results against ICXML (Zhu and Zamani, 2024) (only LLM baseline for EZ-XMC) using various LLM models as shown in Table 3. On EURLex-4K, LMTX significantly outperforms all ICXML variants, achieving a P@1 of 47.28 versus 19.14. On LF-WikiSeeAlso-320K, LMTX demonstrates superior performance compared to models up to 33B in size, with an insignificant performance drop relative to the substantial difference in model size (70B vs 66M). Crucially, LMTX achieves these results with significantly reduced computational demands and substantially lower inference times, enabling more scalable real-world deployment.

5 Ablations and Comprehensive Analysis

Evaluating Teacher Models (Analyzing Open-Source LLMs): We evaluate open-source LLMs as potential teacher models. Table 4 presents the performance results using different recently released LLM model families, with same parameters (13B).

Our analysis reveals that WizardLM outperforms other models on the AmazonCat-13K and LF-WikiSeeAlso-320K datasets, while Llama2 demonstrates improved performance over WizardLM on the LF-Wikipedia-500K dataset. These findings

[3]https://github.com/huiyegit/APLC_XLNet

[4]http://manikvarma.org/downloads/XC/XMLRepository.html

Method	P@1	P@3	P@5	R@1	R@3	R@5	R@10	P@1	P@3	P@5	R@1	R@3	R@5	R@10
	EURLex-4K							**Wiki10-31K**						
Glove	1.66	1.11	1.04	0.37	0.73	1.08	1.88	3.87	3.11	2.87	0.24	0.57	0.89	1.48
SentBERT	8.52	7.70	6.83	1.70	4.54	6.69	10.20	9.39	6.93	5.81	0.60	1.31	1.81	2.70
SimCSE	5.86	4.44	3.85	1.20	2.86	3.93	6.12	23.55	17.21	14.01	1.42	3.07	4.13	6.01
MPNet	10.81	8.65	7.21	2.27	5.28	7.28	10.85	44.82	29.18	22.38	2.63	5.12	6.52	8.89
Msmacro-distilbert	15.91	9.89	7.81	3.33	6.16	8.08	11.22	54.17§	33.44§	25.38§	3.18§	5.82§	7.32§	9.70§
RTS	30.58§	21.54§	17.73§	6.19§	13.01§	17.72§	25.34§	47.73	31.03	23.65	2.81	5.41	6.84	9.12
LMTX	**47.28**†	**29.34**†	**21.98**†	**9.6**†	**17.68**†	**21.96**†	**28.44**†	**57.89**†	**38.00**†	**29.09**†	**3.41**†	**6.68**†	**8.46**†	**11.14**†
	AmazonCat-13K							**LF-WikiSeeAlso-320K**						
Glove	4.83	3.89	3.42	0.99	2.46	3.67	6.05	3.86	2.76	2.21	2.12	4.11	5.22	6.95
SentBERT	5.21	4.22	3.68	0.99	2.34	3.37	5.35	1.71	1.27	1.06	1.08	2.16	2.90	4.17
SimCSE	2.84	2.60	2.42	0.52	1.41	2.17	3.75	9.03	6.64	5.22	4.99	9.89	12.34	15.93
ICT	15.52	10.48	8.34	2.91	5.93	7.86	11.04	10.76	10.05	8.12	6.12	14.32	18.05	23.01
MPNet	18.01	12.84	10.51	3.63	7.68	10.48	15.73	13.75	11.93	9.58	8.14	17.77	22.21	28.11
MACLR	10.66	6.75	5.14	1.98	3.79	4.81	6.35	16.31	13.53	10.78	9.71	20.39	25.37	32.05
Msmacro-distilbert	16.36	10.96	8.68	3.29	6.62	8.73	12.23	14.93	12.65	10.08	8.99	19.25	23.99	30.19
RTS	18.89§	13.59§	11.07§	3.69§	8.03§	10.97§	16.20§	18.64	**15.14**	**12.07**	10.86	**22.68**	**28.29**	**35.47**
LMTX	**25.91**†	**17.08**†	**13.12**†	**5.53**†	**10.77**†	**13.60**†	**17.84**†	**19.11**	14.00	10.95	**11.41**	21.38	26.10	32.44
	LF-Wikipedia-500K													
Glove	2.19	1.52	1.23	0.85	1.66	2.18	3.10							
SentBERT	0.17	0.15	0.13	0.05	0.13	0.18	0.30							
SimCSE	14.32	6.84	4.55	4.24	8.03	11.26	14.35							
ICT	17.74	9.67	7.06	7.35	11.60	13.84	17.19							
MPNet	22.46	12.87	9.49	8.74	14.07	16.76	20.64							
Msmacro-distilbert	21.62	12.75	9.52	8.27	13.81	16.68	20.89							
MACLR	28.44	17.75	13.53	10.40	18.16	22.38	28.52							
RTS	30.67	19.03	14.34	10.58	18.48	22.51	28.23							
LMTX	**40.25**	**23.00**	**16.81**	**13.65**	**22.15**	**26.16**	**31.61**							

Table 2: Comparison of LMTX model with state-of-the-art EZ-XMC methods. The symbol † indicates a statistically significant improvement over the best baseline model (paired t-test with $p \leq 0.01$) and the symbol § represents the best baseline model.

underscore the versatility of our proposed method, which is not confined to a single LLM model. This flexibility enables selection of the most appropriate teacher model to achieve optimal performance.

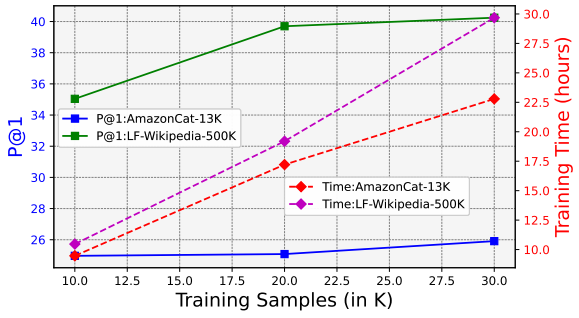

Figure 3: Effect of training sample size on LMTX performance and training time.

Optimizing Training Efficiency: Impact of Sample Size on Performance and Training Time To enhance the efficiency and cost-effectiveness of LMTX, particularly when incorporating LLM-based teacher models in large datasets, we trained the bi-encoder using a subset of the training dataset. As depicted in Figure 3, we systematically investigated the impact of reducing the number of documents on both final performance and training time by randomly sampling data from the entire dataset. The results show that increasing the number of training samples improves model performance, as evidenced by higher P@1 scores. However, this improvement is accompanied by a significant increase in training time, underscoring the necessity of balancing performance gains with the corresponding training time.

Assessing Initialization Robustness: The choice of initialization influences both the quality of the initial label shortlist and the subsequent training process of the bi-encoder. To isolate the effects of our method from potential biases due to initialization advantages, we applied identical initialization procedures to both our approach and the best non-LLM baseline RTS (Zhang et al., 2022). The results, as presented in Table 5, demonstrate that our method consistently outperforms the baseline, even when identical initialization is applied. We also include the performance of the initialized bi-encoder model, msmarco-distilbert-base-v4[5], in Ta-

[5]https://huggingface.co/sentence-transformers/msmarco-distilbert-base-v4

Dataset	Methods	P@1	P@5	R@1	R@5	Inf. time	GPUs
EURLex-4K	ICXML-WizardLM-13B	2.21	2.28	0.5	2.39	16.46	1x(A100-40GB)
	ICXML-Vicuna-33B	7.47	6.05	1.64	6.15	35.28	2x(A100-40GB)
	ICXML-Llama3-70B	19.14	16.51	3.85	16.27	21.32	4x(A100-80GB)
	LMTX-DistilBERT-66M	**47.28**	**21.98**	**9.6**	**21.96**	0.019	1x(A100-40GB)
LF-WikiSeeAlso-320K	ICXML-WizardLM-13B	4.38	2.94	2.74	8.28	19.48	1x(A100-40GB)
	ICXML-Vicuna-33B	6.6	4.12	3.76	11.29	24.29	2x(A100-40GB)
	ICXML-Llama3-70B	**26.13**	**13.93**	**13.54**	**31.02**	15.53	4x(A100-80GB)
	LMTX-DistilBERT-66M	19.22	10.94	11.15	26.01	0.032	1x(A100-40GB)

Table 3: Performance comparison of LMTX and ICXML on EURLex-4K and LF-WikiSeeAlso-320K datasets. The table shows precision and recall metrics, inference time (in hours), and the number of GPUs used. Results for LF-WikiSeeAlso-320K are averaged over two 3500-sample subsets.

Dataset	LLM Model	P@1	P@5	R@1	R@5
AmazonCat-13K	WizardLM	**25.91**	**13.12**	**5.53**	**13.60**
	Vicuna	25.01	12.70	5.24	12.95
	Llama2	25.21	12.76	5.34	13.22
LF WikiSeeAlso-320K	WizardLM	**19.11**	10.95	**11.41**	26.10
	Vicuna	17.76	**11.07**	10.91	**26.58**
	Llama2	17.59	10.64	10.46	25.27
LF Wikipedia-500K	WizardLM	40.25	16.81	13.65	26.16
	Vicuna	39.37	16.78	13.47	26.04
	Llama2	**41.67**	**17.20**	**14.37**	**26.86**

Table 4: Comparison of different LLM models as teacher.

Dataset	Initialization	P@1	P@5	R@1	R@5
AmazonCat-13K	RTS-SI	17.87	10.35	3.57	10.61
	LMTX	**25.91**	**13.12**	**5.53**	**13.60**
LF- WikiSee Also-320K	RTS-SI	14.82	8.89	8.41	21.02
	LMTX	**19.11**	**10.95**	**11.41**	**26.10**

Table 5: Performance comparison across datasets with consistent initialization. RTS-SI uses same initialization as ours.

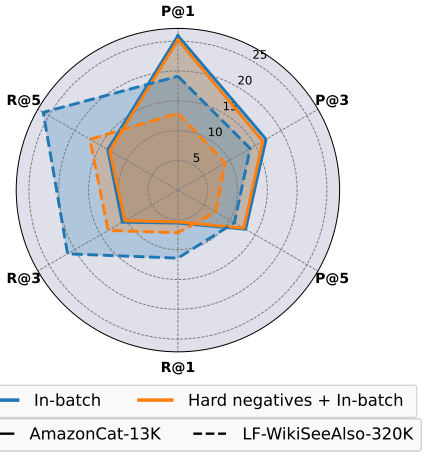

Figure 4: Comparative impact of negative sampling strategies on precision and recall performance.

ble 2. The results demonstrate that training with the proposed method improves the bi-encoder model, making it outperform the initialized model on XMC problems. These results indicate that our method's efficacy stems from intrinsic improvements in the learning process rather than initialization advantages, underscoring its robustness and broad applicability.

Evaluating Negative Sampling and the Impact of LLM-Derived Hard Negatives: Our bi-encoder training employs in-batch negatives. We extended this approach by incorporating hard negatives, identified by the LLM model and tagged as "no". For each document, we constructed a negative set comprising these hard negatives and the pseudo-positive labels of other documents within the same batch. Figure 4 illustrates the comparative performance of these strategies. Notably, our results indicate that the inclusion of hard negatives can potentially impede bi-encoder training, likely due to the risk of introducing false negatives.

6 Related Work

Supervised Extreme Multi-label Text Classification : Supervised XMC methods leveraging non-label features include one-vs-rest approaches (Yen et al., 2016; Babbar and Schölkopf, 2017, 2019; Schultheis and Babbar, 2022), which are based on TF-IDF representations, as well as tree-based methods (You et al., 2019; Yu et al., 2022; Chang et al., 2020; Jiang et al., 2021; Liu et al., 2021; Gupta et al., 2022; Zhang et al., 2021; Kharbanda et al., 2022; Khandagale et al., 2020) that train distinct classifiers for different levels of the tree. State-of-the-art non-label feature methods (Kharbanda et al., 2022; Zhang et al., 2021; Jiang et al., 2021) are based on a transformer encoder and multi-layered tree classifiers. In contrast, state-of-the-art label feature methods (Saini et al., 2021; Dahiya et al., 2021; Mittal et al., 2021b; Dahiya et al., 2023; Mit-

tal et al., 2021a; Gupta et al., 2023) focus on embedding both label text and document text to achieve high accuracy. All of these supervised approaches rely on well-annotated datasets and require comprehensive coverage of most of the labels in the training dataset.

Zero-shot Extreme Multi-label Text Classification: The zero-shot XMC is capable of handling unseen labels which are not in the training dataset. ZestXML (Gupta et al., 2021) applied generated TF-IDF features and a trained linear model to enable retrieval of unseen labels. However, this method still relies on well-annotated training datasets to learn the linear model and is not suitable for the cold start scenario. Another extreme setting in zero-shot XMC is Extreme Zero-shot Extreme Multi-label Text Classification (EZ-XMC) (Xiong et al., 2022). EZ-XMC is specifically designed for the zero-shot scenario, particularly tailored for the cold start scenario without the need for a well-annotated training dataset. The key distinction between zero-shot XMC and EZ-XMC lies in whether annotated labels are employed in the training process. Unlike zero-shot XMC, EZ-XMC does not utilize any annotated labels. We adopt the EZ-XMC setting in this paper. MACLR (Xiong et al., 2022) proposes a multi-stage self-supervised approach for EZ-XMC by using pseudo pairs of (title, document). On the other hand, RTS (Zhang et al., 2022) introduces a randomized text segmentation method to construct pseudo positive labels with segments within one document.

Dense Sentence Embedding: In the domains of open domain question answering and information retrieval, ICT (Inverse Cloze Task) (Lee et al., 2019) constructs positive passages by extracting random sentences and their corresponding contexts from the documents. MSS(Guu et al., 2020) shows that the ICT encoder can be improved by predicting the masked salient spans with a reader. Spider (Ram et al., 2022) adopts sentences that contain recurring spans as positive passage. Both HLP (Zhou et al., 2022) and WLP(Chang et al., 2019) utilize hyperlinks within Wikipedia pages to construct positive passages. ART (Sachan et al., 2023) tries to guide the training of bi-encoder via the question reconstruction score. Additionally, there are works that focus on sentence similarity, including (i) SimCSE (Gao et al., 2021) introduces a contrastive learning framework that employs dropout noise as augmented positives, and (ii) Sentence-BERT (Reimers and Gurevych, 2019) introduces a supervised siamese transformer framework.

Large Language Models XMC Applications: LLM models such as GPT-3 (Brown et al., 2020), and GPT-4 (OpenAI, 2023) have demonstrated their zero-shot effectiveness in various NLP downstream tasks. In XMC, Xu et al. (2023b) employed LLM to construct a thesaurus for labels in a few-shot setting. Liu et al. (2024) applied LLM for incremental XMC setting. Zhu and Zamani (2024), on the other hand, directly applied the LLM for inference in EZ-XMC setting. This approach predominantly focuses on recommendation datasets and relies on the costly GPT-3.5 and GPT-4 for inference. In contrast, our methods concentrate on tagging tasks and emphasize swift inference through a lightweight bi-encoder.

7 Conclusion

This paper introduces a novel approach to address the EZ-XMC tagging and categorization challenge. We leverage an LLM as a teacher to guide the training of the bi-encoder model. Unlike existing methods, our approach effectively handles the issue of low-quality training pairs. Additionally, our algorithm enables faster inference without the need for an LLM during prediction, providing a significant advantage over LLM-based methods and supporting lightweight deployment in EZ-XMC scenarios. Performance evaluations demonstrate that our method achieves state-of-the-art results across multiple datasets. Ablation experiments further highlight its potential for improved performance when using alternative teacher models. For future work, exploring more efficient ways to integrate the LLM model is interesting, such as transitioning from point-wise to list-wise prompts, could be an exciting direction.

8 Limitations

While our method demonstrates superior performance with a smaller subset, there is potential for further improvements with a larger training set (Figure 3). However, our current LLM pseudo-labeling approach relies on point-wise feedback, which is time-consuming. For the comparison with ICXML, we employed publicly available open-source models instead of GPT-3.5, which is specified in the original ICXML implementation. Despite this, benchmarking on large-label datasets proved computationally prohibitive. Instead, we used a subset of the test set and repeated the experiments multiple

times to ensure statistical significance.

Acknowledgements

We thank reviewers for their valuable comments and suggestions, we also sincerely thank Ansh Arora for his assistance in evaluating certain baselines of Zero-shot XMC models. We acknowledge the support of Research Council of Finland (Academy of Finland) via grants 347707 and 348215. We also thank the Aalto Science-IT project, and CSC IT Center for Science, Finland for the computational resources provided.

References

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

Rohit Babbar and Bernhard Schölkopf. 2019. Data scarcity, robustness and extreme multi-label classification. *Machine Learning*, 108(8-9):1329–1351.

K. Bhatia, K. Dahiya, H. Jain, P. Kar, A. Mittal, Y. Prabhu, and M. Varma. 2016. The extreme classification repository: Multi-label datasets and code.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Data augmentation for information retrieval using large language models. *arXiv preprint arXiv:2202.05144*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Wei-Cheng Chang, X Yu Felix, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2019. Pre-training tasks for embedding-based large-scale retrieval. In *International Conference on Learning Representations*.

Wei-Cheng Chang, Daniel Jiang, Hsiang-Fu Yu, Choon Hui Teo, Jiong Zhang, Kai Zhong, Kedarnath Kolluri, Qie Hu, Nikhil Shandilya, Vyacheslav Ievgrafov, et al. 2021a. Extreme multi-label learning for semantic matching in product search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2643–2651.

Wei-Cheng Chang, Daniel L. 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. 2021b. Extreme multi-label learning for semantic matching in product search. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 2643–2651. ACM.

Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S Dhillon. 2020. Taming pretrained transformers for extreme multi-label text classification. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3163–3171.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Kunal Dahiya, Ananye Agarwal, Deepak Saini, K Gururaj, Jian Jiao, Amit Singh, Sumeet Agarwal, Purushottam Kar, and Manik Varma. 2021. Siamesexml: Siamese networks meet extreme classifiers with 100m labels. In *International Conference on Machine Learning*, pages 2330–2340. PMLR.

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

Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering chatgpt's capabilities in recommender systems. *arXiv preprint arXiv:2305.02182*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Nilesh Gupta, Sakina Bohra, Yashoteja Prabhu, Saurabh Purohit, and Manik Varma. 2021. Generalized zero-shot extreme multi-label learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 527–535.

Nilesh Gupta, Patrick Chen, Hsiang-Fu Yu, Cho-Jui Hsieh, and Inderjit Dhillon. 2022. Elias: End-to-end learning to index and search in large output spaces. *Advances in Neural Information Processing Systems*, 35:19798–19809.

Nilesh Gupta, Fnu Devvrit, Ankit Singh Rawat, Srinadh Bhojanapalli, Prateek Jain, and Inderjit S Dhillon. 2023. Efficacy of dual-encoders for extreme multi-label classification. In *The Twelfth International Conference on Learning Representations*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2023. Large language models are zero-shot rankers for recommender systems. *arXiv preprint arXiv:2305.08845*.

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

Ting Jiang, Deqing Wang, Leilei Sun, Huayi Yang, Zhengyang Zhao, and Fuzhen Zhuang. 2021. Lightxml: Transformer with dynamic negative sampling for high-performance extreme multi-label text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7987–7994.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

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

Siddhant Kharbanda, Atmadeep Banerjee, Erik Schultheis, and Rohit Babbar. 2022. Cascadexml: Rethinking transformers for end-to-end multi-resolution training in extreme multi-label classification. *Advances in Neural Information Processing Systems*, 35:2074–2087.

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

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pages 115–124.

Xuanqing Liu, Wei-Cheng Chang, Hsiang-Fu Yu, Cho-Jui Hsieh, and Inderjit Dhillon. 2021. Label disentanglement in partition-based extreme multilabel classification. *Advances in Neural Information Processing Systems*, 34:15359–15369.

Yanjiang Liu, Tianyun Zhong, Yaojie Lu, Hongyu Lin, Ben He, Shuheng Zhou, Huijia Zhu, Weiqiang Wang, Zhongyi Liu, Xianpei Han, et al. 2024. Xmc-agent: Dynamic navigation over scalable hierarchical index for incremental extreme multi-label classification. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 5659–5672.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document reranking with a large language model. *arXiv preprint arXiv:2305.02156*.

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

R. Manmatha, Chaoxia Wu, Alex Smola, and Philipp Krähenbühl. 2017. Sampling matters in deep embedding learning. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2859–2867.

Anshul Mittal, Kunal Dahiya, Sheshansh Agrawal, Deepak Saini, Sumeet Agarwal, Purushottam Kar, and Manik Varma. 2021a. Decaf: Deep extreme classification with label features. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 49–57.

Anshul Mittal, Noveen Sachdeva, Sheshansh Agrawal, Sumeet Agarwal, Purushottam Kar, and Manik Varma. 2021b. Eclare: Extreme classification with label graph correlations. In *Proceedings of the Web Conference 2021*, pages 3721–3732.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Mohammadreza Qaraei, Erik Schultheis, Priyanshu Gupta, and Rohit Babbar. 2021. Convex surrogates for unbiased loss functions in extreme classification with missing labels. In *Proceedings of the Web Conference 2021*, pages 3711–3720.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, et al. 2023. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563*.

Ori Ram, Gal Shachaf, Omer Levy, Jonathan Berant, and Amir Globerson. 2022. Learning to retrieve passages without supervision. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2687–2700.

Sashank J. Reddi, Satyen Kale, Felix Yu, Daniel Holtmann-Rice, Jiecao Chen, and Sanjiv Kumar. 2019. Stochastic negative mining for learning with large output spaces. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1940–1949. PMLR.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Nils Reimers and Iryna Gurevych. 2021. The curse of dense low-dimensional information retrieval for large index sizes. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 605–611.

Jon Saad-Falcon, Omar Khattab, Keshav Santhanam, Radu Florian, Martin Franz, Salim Roukos, Avirup Sil, Md Arafat Sultan, and Christopher Potts. 2023. Udapdr: Unsupervised domain adaptation via llm prompting and distillation of rerankers. *arXiv preprint arXiv:2303.00807*.

Devendra Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving passage retrieval with zero-shot question generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3781–3797, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Deepak Saini, Arnav Kumar Jain, Kushal Dave, Jian Jiao, Amit Singh, Ruofei Zhang, and Manik Varma. 2021. Galaxc: Graph neural networks with labelwise attention for extreme classification. In *Proceedings of the Web Conference 2021*, pages 3733–3744.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015a. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015b. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Erik Schultheis and Rohit Babbar. 2021. Unbiased loss functions for multilabel classification with missing labels. *arXiv preprint arXiv:2109.11282*.

Erik Schultheis and Rohit Babbar. 2022. Speeding-up one-versus-all training for extreme classification via mean-separating initialization. *Machine Learning*, 111(11):3953–3976.

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

Erik Schultheis, Marek Wydmuch, Wojciech Kotlowski, Rohit Babbar, and Krzysztof Dembczynski. 2024. Generalized test utilities for long-tail performance in extreme multi-label classification. *Advances in Neural Information Processing Systems*, 36.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agent. *arXiv preprint arXiv:2304.09542*.

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

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

Yuanhao Xiong, Wei-Cheng Chang, Cho-Jui Hsieh, Hsiang-Fu Yu, and Inderjit Dhillon. 2022. Extreme Zero-Shot learning for extreme text classification. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5455–5468, Seattle, United States. Association for Computational Linguistics.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023a. Wizardlm: Empowering large language models to follow complex instructions. *Preprint*, arXiv:2304.12244.

Nan Xu, Fei Wang, Mingtao Dong, and Muhao Chen. 2023b. Dense retrieval as indirect supervision for large-space decision making. *arXiv preprint arXiv:2310.18619*.

Ian En-Hsu Yen, Xiangru Huang, Pradeep Ravikumar, Kai Zhong, and Inderjit Dhillon. 2016. Pd-sparse: A primal and dual sparse approach to extreme multiclass and multilabel classification. In *International conference on machine learning*, pages 3069–3077. PMLR.

Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2019. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Hsiang-Fu Yu, Kai Zhong, Jiong Zhang, Wei-Cheng Chang, and Inderjit S Dhillon. 2022. Pecos: Prediction for enormous and correlated output spaces. *the Journal of machine Learning research*, 23(1):4233–4264.

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

Tianyi Zhang, Zhaozhuo Xu, Tharun Medini, and Anshumali Shrivastava. 2022. Structural contrastive representation learning for zero-shot multi-label text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4937–4947, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jiawei Zhou, Xiaoguang Li, Lifeng Shang, Lan Luo, Ke Zhan, Enrui Hu, Xinyu Zhang, Hao Jiang, Zhao Cao, Fan Yu, et al. 2022. Hyperlink-induced pretraining for passage retrieval in open-domain question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7135–7146.

Yaxin Zhu and Hamed Zamani. 2024. Icxml: An in-context learning framework for zero-shot extreme multi-label classification. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2086–2098.

A Appendix

A.1 Implementation Details

Bi-Encoder: In our bi-encoder framework, we adopt a siamese network architecture for sentence encoding. The core of this network is DistilBERT (Sanh et al., 2019), comprising six transformer layers. For the generation of sentence embeddings, we apply mean pooling, yielding embeddings of 768 dimensions. The bi-encoder is initialized using the msmarco-distilbert-base-v4[6], and ANNs is built via the HNSW package[7]. For optimization, we employ the AdamW optimizer (Loshchilov and Hutter, 2018) with a learning rate of 0.0002, setting the batch size to 128. All experiments for training the bi-encoder are conducted on a single A100 GPU. Following the supervised method in (Dahiya et al., 2023), we have used triplet loss (Schroff et al., 2015b; Liu et al., 2017) with margin γ is set to 0.3. For model selection, a development set of 800 documents is randomly selected from the training dataset, with pseudo labels derived from the top-k labels as determined by the LLM model.

LLM: For our Large Language Model (LLM) component, we employ the WizardLM-13B-V1.0 model (Xu et al., 2023a), an open-source LLM notable for achieving 89.1% of GPT-4's (OpenAI, 2023) performance with approximately 13 billion parameters. In addition, for the purposes of this study, we incorporate Llama2 (Touvron et al., 2023) and vicuna-13b-v1.3 (Chiang et al., 2023) models in our ablation experiments to serve as comparative benchmarks. All LLM computations are performed on 2 $\times$ A100 GPUs, with input instances truncated to 430 tokens. For the comparison with ICXML (Zhu and Zamani, 2024), we adopt Llama3 (Dubey et al., 2024) and vicuna-33b-v1.3 (Chiang et al., 2023) for inference.

Random Training Subsets: To minimize bias from random subsets for AmazonCat-13K, LF-WikiSeeAlso-320K, and LF-Wikipedia-500K, we conducted three separate random samplings and used the average performance of the three models on the test set as our final result in Table 2.

A.2 Evaluation Metrics

We employ the commonly used evaluation metrics (Reddi et al., 2019; Chang et al., 2021b; Zhang et al., 2022) for the EZ-XMC setting : $Precision@k(P@m)$ and $Recall@m(R@m)$.

$$P@m = \frac{1}{m} \sum_{i \in rank_m(\hat{y})} y_i, \quad R@m = \frac{1}{\sum_l y_l} \sum_{i \in rank_m(\hat{y})} y_i \tag{2}$$

where $\hat{y} \in \mathbb{R}^L$ represents a vector containing the predicted labels' score for each instance, while $y \in \{0,1\}^L$ corresponds to a vector representing the ground truth for each document. The term $rank_m(\hat{y})$ refers to a list of the predicted top-m

[6]https://huggingface.co/sentence-transformers/msmarco-distilbert-base-v4

[7]https://github.com/kunaldahiya/pyxclib

label indices. The definition of the two metrics applies to a single instance; for multiple instances, the performance is the average across all instances.

Dataset	Models	Training	GPUs
AmazonCat-13K	MACLR	28.86	4 A100
	RTS	35.60	4 A100
	LMTX 30k	**22.79**	2 A100
LF-WikiSeeAlso-320K	MACLR	28.88	4 A100
	RTS	26.66	4 A100
	LMTX 30k	**26.03**	2 A100

Table 6: The training time (in hours) comparison with non-LLM methods.

A.3 Training time

In Table 6, we present the training time for our model when trained with a subset of the training set. The table shows that LMTX's time efficiency is competitive or even superior compared to other models, especially in the context of larger datasets. These results underscore the effectiveness of LMTX, even with the incorporation of the LLM model.

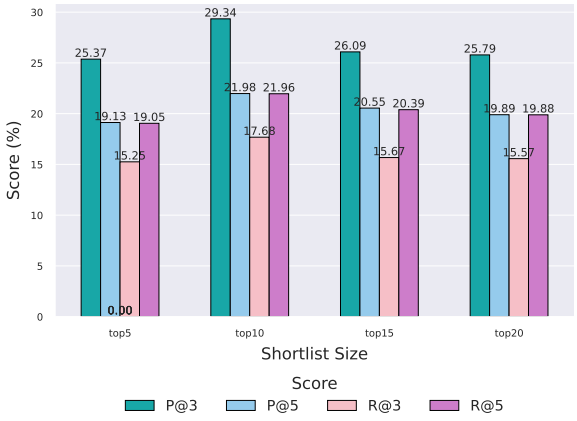

Figure 5: Impact of shortlist size on performance metrics for EURLex-4k dataset.

A.4 Sensitivity to the Shortlist Size

The size of the shortlist directly impacts both the quality of pseudo labels generated by the LLM and the computational efficiency of the screening process. We empirically evaluated the effect of varying shortlist sizes on precision and recall for the EURLex-4K dataset, as illustrated in Figure 5. Our results demonstrate that while a shortlist size of 5 negatively impacts performance, increasing the size beyond 10 does not yield significant improvements. Notably, we observed optimal performance across multiple metrics at a shortlist size of 10, indicating that our approach achieves superior results with a relatively compact shortlist, thereby enhancing training efficiency.

A.5 Evaluating Pseudo-Label Quality and the Role of Curriculum Learning:

To assess the LLM's capability in selecting relevant labels and the quality of the selected pseudo-labels, we measured the overlap between the pseudo-labels and the supervised ground truth. The overlap ratio is calculated as follows

$$quality = \frac{len(pseudo\_labels \cap true\_labels)}{len(true\_labels)}$$

As illustrated in Figure 6, the overlap ratio progressively increases across training epochs for both the EURLex-4K and AmazonCat-13K datasets. This trend demonstrates the effectiveness of our curriculum learning framework, as the LLM refines its label selection over time, resulting in higher-quality pseudo-labels. The increasing overlap highlights that the curriculum learning strategy not only improves pseudo-label alignment with ground truth but also enhances the performance of the bi-encoder.

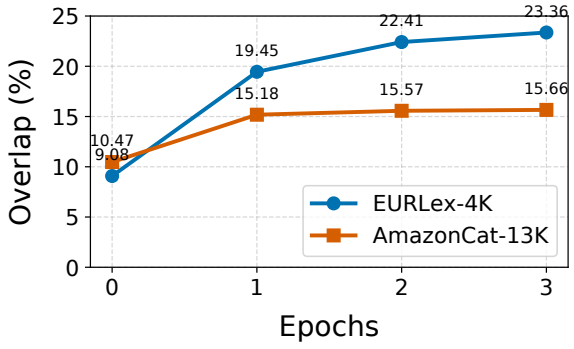

Figure 6: Overlap ratio between LLM-generated pseudo-labels and ground truth labels across training epochs for EURLex-4K and AmazonCat-13K datasets.

A.6 Prompts for LLM

- **EURLex-4k** and **Wiki10-31K**: "document = {doc}. Is the tag {label_text} relevant to the document? answer yes or no"
- **AmazonCat-13K**: "document = {doc}. The document is amazon product description, Is the tag {label_text} relevant to the document? answer yes or no"

- **LF-WikiSeeAlso-320K**: "document = {doc}. The document is the wikipedia page. Does another wikipedia page name "{label_text}" has the relation to the document? answer yes or no"
- **LF-Wikipedia-500K**:"document = {doc}, the document is the wikipedia page. Is the tag "{label_text}" relevant to the document? answer yes or no".