ICXML: An In-Context Learning Framework for Zero-Shot Extreme Multi-Label Classification

Anonymous ACL submission

Abstract

This paper focuses on the task of Extreme Multi-Label Classification (XMC) whose goal is to predict multiple labels for each instance from an extremely large label space. While existing research has primarily focused on fully supervised XMC, real-world scenarios often lack complete supervision signals, highlighting the importance of zero-shot settings. Given the large label space, utilizing in-context learning approaches is not trivial. We address this issue by introducing In-Context Extreme Multi-label Learning (ICXML), a two-stage framework that cuts down the search space by generating a set of candidate labels through in-context learning and then reranks them. Extensive experiments suggest that ICXML advances the state of the art on two diverse public benchmarks.

1 Introduction

Extreme Multi-Label Classification (XMC) deals with the classification of instances into a set of relevant labels from a large label set (Bhatia et al., 2015; Mittal et al., 2021; Dahiya et al., 2023). It finds applications in various domains, including text categorization (Chalkidis et al., 2019), recommendation systems (Agrawal et al., 2013), image tagging (Mittal et al., 2022), and so on. Unlike conventional multi-label classification, where the number of labels is relatively small, XMC involves an exponentially larger label space, e.g., in the 10^6 magnitude. This poses significant computational and modeling challenges.

While existing research has primarily focused on supervised XMC, real-world applications often encounter challenges in obtaining complete supervision signals. Scenarios arise during test sessions when new labels emerge without any assigned input instances (Gupta et al., 2021), or when both instances and labels are available, but the corresponding relations between them are unknown (Xiong et al., 2021). This task, which is the focus of this paper, is called zero-shot XMC.

Zero-shot XMC can be seen as a *retrieval* problem, where the test instance is considered as the query and candidate labels are retrieved in response to the given input. Methods based on lexical matching, such as TF-IDF (Salton and Buckley, 1988) and BM25 (Robertson et al., 1995), and semantic matching, such as dense retrieval models (Hofstätter et al., 2021; Karpukhin et al., 2020), can be adopted for this task. State-of-the-art approaches for zero-shot XMC, such as RTS (Zhang et al., 2022a) and MACLR (Xiong et al., 2021), also belong to this category. A major shortcoming with these approaches is that there is little lexical or semantic overlap between the test instance (i.e., queries) and the label space (i.e., documents). One may argue that large language models (LLMs) can be used to generate labels for each test input. However, the labels that LLMs generate may not be in the acceptable label set and unlike conventional classification tasks, the label set is too large to be given to the LLMs in their prompt. Therefore, using LLMs for this task is either impractical or extremely expensive.

In this work, we put together the benefits of both retrieval- and generation-based approaches by introducing ICXML– a two-stage framework designed for zero-shot XMC. In the first stage - generate, ICXML enriches the intrinsic capabilities of large language models by generating and/or retrieving demonstrations in a zero-shot manner. The obtained outputs are generated by prompting the model with the help of a support set of generated demonstrations. Subsequently, the generated outputs are adjusted to align with the label space, resulting in a condensed shortlist of candidate labels. In the second stage - rerank, we leverage the capabilities of LLMs to perform multi-label classification by reintroducing the refined candidate label shortlist along with the test instance as input.

This approach capitalizes on the language model's inherent potential to handle multiple labels concurrently, thereby augmenting its performance in the context of extreme multi-label classification tasks.

In summary, our main contributions are three fold:

1. Introducing a two-stage framework for zero-shot XMC, involving generation-based label shortlisting and label reranking.
2. Advocating for a generation-based approach to yield high-quality input-label pairs instead of retrieval-based. This method also addresses the challenges posed by the absence of specific input scenarios, ensuring robustness across diverse contexts.
3. Advancing state of the art in zero-shot XMC on two public benchmarks, i.e., LF-Amazon-131K and LF-WikiSeeAlso-320K, and providing detailed analysis for a deeper understanding of model performance. We show that ICXML performs effectively even without reliance on an input corpus – a collection of input candidates that is used by state-of-the-art baselines (Xiong et al., 2021; Zhang et al., 2022a).

2 Related Work

2.1 Extreme Multi-label Classification

Extreme classification refers to the task of making predictions over vast label spaces, typically comprising thousands to millions of classes, with multiple correct classes assigned to each instance (Agrawal et al., 2013; Bhatia et al., 2015; Liu et al., 2017; Jiang et al., 2021; Dahiya et al., 2021; Mittal et al., 2021; Dahiya et al., 2023). In this context, (Gupta et al., 2021) framework focuses on predicting unseen labels, while Zhang et al. (2022b) handles instances where no labels are observed. Prior work by Simig et al. (2022) explores the generation of labels usingLLMs for this specific task. Additionally, Xiong et al. (2021) investigated a generalized zero-shot setting where no annotations are available. These research endeavors contribute to the advancement and understanding of extreme classification, addressing challenges related to unseen labels, missing label information, and generalized zero-shot scenarios (Zhang et al., 2022a; Aggarwal et al., 2023). The zero-shot setting has found applicability in various real-world scenarios including cold start recommendation tasks, and is mainly solved with dependency on large-scale training by creating pseudo annotations. In our work, we propose a fully zero-shot setting and aim to tackle it through the utilization of in-context learning.

2.2 In-Context Learning

The scaling of model size and corpus size has led to notable advancements in LLMs (Brown et al., 2020; Chowdhery et al., 2022), enabling them to demonstrate remarkable ICL capabilities (Wei et al., 2022a). These models have showcased their ability to effectively learn from a limited number of examples provided within the context. The research community has witnessed the emergence of numerous studies focusing on the analysis and enhancement of demonstrations in ICL (Wei et al., 2022b; Fu et al., 2022). (Liu et al., 2021; Rubin et al., 2021; Luo et al., 2023; Chen et al., 2022; Ram et al., 2023; Cheng et al., 2023) explored the retrieval of influential demonstrations from a training corpus to provide effective guidance. Additionally, (Lyu et al., 2022; Chen et al., 2023) proposed the generation of pseudo demonstrations to tackle zero-shot scenarios. In addition, a plenty of studies have focused on investigating the ranking ability of models in various tasks characterized by a vast search space. These tasks encompass areas such as information retrieval (Shen et al., 2023; Gao et al., 2022), reranking (Sun et al., 2023; Ma et al., 2023), and recommendation systems(Hou et al., 2023). Motivated by these advancements, our objective is to identify an optimal ICL method suitable for extreme multi-label classification, considering the specific requirements and challenges of this task. Through our research, we aim to contribute to the development of effective and efficient ICL techniques that can address the complexities of extreme multi-label classification.

3 Problem Formulation

Let $\mathcal{X}$ and $\mathcal{Y}$ respectively denote the input and output spaces. In this work, we focus on text data, thus each $x \in \mathcal{X}$ is an unstructured text and each $y \in \mathcal{Y}$ is represented by a short text description. Given the focus on XMC, the output space is extremely large, e.g., $|\mathcal{Y}| \sim 10^6$. The goal is to map each input x to a small subset of labels $Y \subset \mathcal{Y}$.

Following Xiong et al. (2021); Zhang et al. (2022a), we also consider a scenario where some input instances, called the input corpus, are available, however the mapping $\{(x_i, Y_i)|x_i \in \mathcal{X}_{\text{train}}, Y_i \subseteq$

$\mathcal{Y}\}$ is not available for training.

4 The ICXML Framework

Benefiting from the high performance and rich knowledge encoded within LLM parameters, we can easily generate a set of labels by describing classification tasks and inputs using prompts. However, for a classification task with a predetermined label set, this approach results in uncertain outcomes in the absence of guidance from few-shot example pairs, referred to as demonstrations (Reynolds and McDonell, 2021; Razeghi et al., 2022). While common approaches involve incorporating the label candidate list into the prompt, this approach becomes impractical or extremely expensive when faced with an extreme label space, e.g., a label space of 10^6 magnitude.

To address this issue, we propose a two-stage framework illustrated by 1. The first stage is generate, including demonstration generation (Section 4.1) and candidate shortlisting (Section 4.2) in the figure. We perform in-context learning using LLM ϕ to generate labels, and the pseudo demonstrations for this stage is generated by ϕ using a prompt-guided approach. The second stage is rerank, where we utilize another prompt-guided method and ϕ for selecting top labels from candidate labels as decribed in Section 4.3.

4.1 Demonstration Generation

For an effective in-context learning performance, it is essential for demonstrations to encompass both the inherent correlation between the input text and the task label, as well as external knowledge that facilitates the model's learning process in relation to the input text. We propose two different strategies to achieve this goal.

Content-based Demonstration Generation: To embody a blend of external knowledge and inherent correlation, the content-based approach first generates relevant and diverse demonstration inputs based on the *test input content*. Each input is then linked to a label that exhibits the inherent correlation.

For each test input $x_i \in \mathcal{X}$, the LLM Φ is employed to generate a set of m demonstration inputs, denoted as $Z_i = \{z_i^1, z_i^2, \cdots, z_i^m\}$, where $z_i^j \sim \Phi(\textsc{Prompt}(x_i, t_1))$, t_1 is the corresponding task description,[1] and p denotes a prompt for Φ. For each $z_i^j \in Z_i$ and each label $l \in \mathcal{Y}$, the zero-shot retriever θ computes a score function as follows: $\text{score}(l) = \theta(l|z_i^j)$. The top n labels, denoted by $L_i^j = \arg\text{top}n\{\text{score}(l)|l \in \mathcal{Y}\}$ are selected based on their scores to form the demonstration set:

$$D_i = \{(z_i^1, L_i^1), (z_i^2, L_i^2), \cdots, (z_i^m, L_i^m)\}. \quad (1)$$

Label-centric Demonstration Generation: The label-centric approach pursues an inherent-external trajectory by first mapping the test input to the label space, capturing labels with high correlation. It then generates demonstration inputs based on these labels.

For each test instance $x_i \in \mathcal{X}$, the zero-shot retriever θ identifies the top n labels, denoted by $L_i = \{l_i^1, l_i^2, \ldots, l_i^n\}$, where $L_i = \arg\text{top}n\{\theta(l|x_i)|l \in \mathcal{Y}\}$. For each label $l_i^j \in L_i$, a pseudo input text is generated: $z_i^j \sim \Phi(p(l_i^j, t_2))$. When duplicate input texts arise, the labels are merged into a label list, denoted by $\mathbf{l}_i^j$. To be consistent with content-based method, let $L_i^j = \{l_i^j\}$, the demonstration set is then constructed as follows:

$$D_i = \{(z_i^1, L_i^1), (z_i^2, L_i^2), \ldots, (z_i^m, L_i^m)\}, \quad (2)$$

where m is the final size of the grouped label list, and l_i^j corresponds to the duplicate input z_i^j.

4.2 Candidate Label Shortlisting

ICL-based Inference After the pseudo demonstration sets D_is are constructed, we integrate them with each test input x_i in the prompt, guiding the few-shot learning process of the language model Φ. Consequently, Φ generates a k-sized set of labels $\hat{Y}_i = \{y_{i,1}, y_{i,2}, \cdots, y_{i,k}\}$, for each $y_{i,j} \in \hat{Y}_i$,

$$y_{i,j} \sim \Phi(\textsc{Prompt}(x_i, D_i, t_3)) \quad (3)$$

where $y_{i,j}$s are the labels produced by the language model, and t_3 denotes the task description. The generation of these labels encapsulates the model's prediction based on both the original input and the pseudo demonstration set.

Label Space Mapping. After inference, we leverage textual semantic matching techniques to establish connections between the generated text and the corresponding labels in the label space. This step is critical for transforming the raw output from the language model into structured labels.

[1] See Appendix A for detailed task descriptions for all 4 different tasks

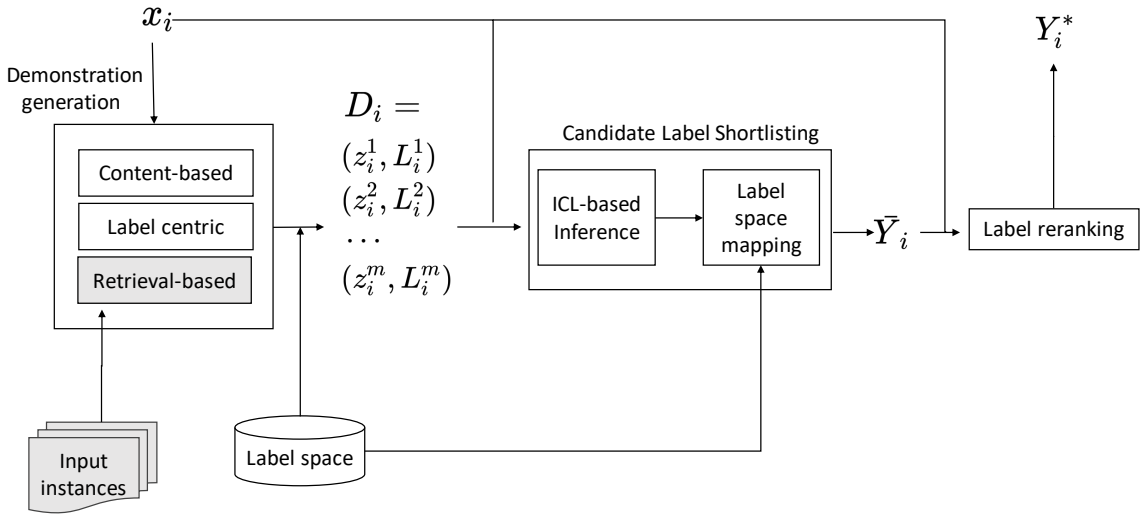

Figure 1: An illustration of the proposed generate-rerank framework. For a given test input x_i, we generate demonstrations D_i to facilitate ICL-based shortlisting. Subsequently, this shortlisted set $\bar{Y}_i$ is provided to LLM for listwise re-ranking, culminating in the final results Y_i^*.

For each generated label $y_{i,j} \in \hat{Y}_i$, we use the zero-shot retriever θ to fetch the top s labels from the label set $\mathcal{Y}$ that possess the highest semantic similarity with $y_{i,j}$. This set, denoted by $\bar{Y}_{i,j}$, is defined as:

$$\bar{Y}_{i,j} = \{\bar{y} = \arg\text{top}s\theta(y|y_{i,j}), y \in \mathcal{Y}\} \quad (4)$$

Finally, we obtain a shortlist

$$\bar{Y}_i = \bigcup_j \bar{Y}_{i,j} \quad (5)$$

for each test instance x_i. Through this process, we map the generated labels to the label space while simultaneously expanding them to a desirable size for in-context learning-based multi-label classification.

4.3 Label Reranking

With the obtained shortlist, our approach effectively contracts the search space for labels, recasting the problem into a standard multi-label classification task. To benefit from this formulation, we feed the whole shortlist into Φ:

$$Y_i^* \sim \Phi(\textsc{Prompt}(x_i, \bar{Y}_i, t_4)), Y_i^* \subseteq \bar{Y}_i \quad (6)$$

Here, a prompt $\textsc{Prompt}(x_i, \bar{Y}_i, t_4)$, steers the LLM to select the most suitable set of labels. The set of labels chosen, denoted by Y_i^*, serves as the final prediction in our approach.

5 Expanding ICXML by Utilizing an Input Corpus

Our methodology adopts a novel approach that avoids the use of training samples, setting it apart from conventional models. This design choice aligns with zero-shot learning paradigms where, although a corpus of training instances $\mathcal{X}_{\text{train}}$ and their corresponding labels may be available, they are not utilized in a paired manner. our framework maintains a degree of adaptability and can be readily extended to a more flexible setting by substituting demonstration generation with demonstration retrieval, when $\mathcal{X}_{\text{train}}$ is available.

For each test input $x_i \in \mathcal{X}$, we select the top m neighbor instances from $\mathcal{X}_{\text{train}}$ as follows:

$$Z_i = \{z|z = \arg\text{top}m\theta(z|x_i), \\ |Z_i| \leq m, z \in \mathcal{X}_{\text{train}}\} \quad (7)$$

Subsequently, for each $z_i^j \in Z_i$, we proceed with the methodology delineated in Section 4.1, titled "Content-based demonstration generation", to construct D_i.

6 Experiments

6.1 Data

We evaluate the effectiveness of our approach on the following large-scale datasets: LF-Amazon-

Dataset	$\lvert X_{train}\rvert$	$\lvert X_{test}\rvert$	$\lvert Y\rvert$
LF-Amazon-131K	294,805	134,835	131,073
LF-WikiSeeAlso-320K	693,082	177,515	312,330

Table 1: Data statistics. The size of training instances, test instances and label space are presented.

131K in item recommendation domain and LF-WikiSeeAlso in Wikipedia articles title tagging domain, where 131K and 320K denote the size of label space (Bhatia et al., 2016). The dataset statistics are presented in Table 1, showcasing the characteristics of each dataset. These benchmark datasets are widely used in evaluation of zero-shot extreme classification settings.

6.2 Baselines

We compare our approach with lexical matching based methods, soft semantic matching based methods, pseudo pretraining based methods and naive zero-shot in-context learning without demonstration augmentation:

Lexical Matching: TF-IDF is a powerful sparse lexical matching technique that matches input tokens to the nearest labels based on similarity in terms of bag-of-words representation (Salton and Buckley, 1988). BM25 is a term-based ranking model that scores documents based on their term frequencies and document lengths. (Robertson et al., 1995).

Soft Matching: The recent development of language models and pre-training + fine-tuning paradigms has paved the way for zero-shot learning using soft matching techniques. Among these models, TAS-B has emerged as an effective and lightweight pre-trained bi-encoder, demonstrating strong generalization capabilities (Hofstätter et al., 2021). TAS-B is trained using dual supervision from a cross-encoder model and ColBERT on the MS MARCO dataset (Nguyen et al., 2016).

MACLR (Xiong et al., 2021) was trained with pseudo positive pairs constructed from TF-IDF.

RTS (Zhang et al., 2022a) proposed a self-supervised auxiliary task for contrastive representation learning that enables end-to-end training.

Free Generation of LLM: The LLM is provided solely with the test input and the task objective, which encompasses elements such as task description, output constraints, among others, to generate a prediction. The prediction is derived by adapting labels from the actual label space (refer to Section 4.2), guided by a heuristic methodology which selects the nearest adaptation in the order of generated raw labels (refer to Section 6.5)

6.3 Experimental Setup

For the LLM Φ, we use OpenAI's GPT-3.5 API in main experiments, while GPT-4 is included in ablation study on a small subset. The LLM was called with a temperature hyperparameter set to 0.0. Instructions are provided in appendix. For semantic matching θ, we use TAS-B model as our semantic macher, so that we can benefit from its knowledge acquired by the pretraining on MS MARCO. For content-based generation, we set $m = 5, n = 5$. For label-centric generation, we set $n = 30$. For inference, we set $k = 10, s = 10$. Following the setup of (Xiong et al., 2021), we use precision and recall as evaluation metrics.

6.4 Main Results

The main experimental results of extreme classification are presented in Table 2. In this table, the various ICXML suffixes denote distinct methods used for constructing demonstrations, and others are the baselines. Here, content-based and label-centric (denoted as "content" and "label") are two different strategies of our method, while demonstration retrieval results (denoted as "retrieval") are recorded to investigate the differential impacts of demonstration retrieval and generation on the in-context learning framework. In general, our method substantially outperforms all baselines when applied to the LF-Amazon-131K dataset, demonstrating a significant enhancement in performance. For LF-WikiSeeAlso-320K dataset, it surpasses the highest performing baseline, soft matching (tas-b), by 3.9% and 1.8% in terms of P@1 and R@1 respectively. However, when evaluating a longer result list, it does not measure up to the performance levels achieved by soft matching. The underperformance might be due to the large label space of 320K, which is significantly larger than 131K. This imposes challenges for the generation-based label space adaptation in terms of fully capturing the label distribution compared to retrieval-based methods. Despite this, we executed experiments on a small subset of the test set using GPT-4 in section 6.5. It was found that this underperformance could be completely mitigated by employing GPT-4 instead of GPT-3.5 for the reranking component of our approach.

It is also observed that for LF-Amazon-131K dataset, the performance of the label-centric ap-

		LF-Amazon-131K							LF-WikiSeeAlso-320K						
	$\mathcal{X}_{\text{train}}$	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
TF-IDF	✗	0.124	0.115	0.091	0.069	0.181	0.231	0.293	0.107	0.089	0.071	0.059	0.130	0.165	0.216
BM25	✗	0.174	0.118	0.088	0.100	0.185	0.224	0.268	0.185	0.120	0.090	0.101	0.175	0.210	0.254
TAS-B	✗	0.135	0.123	0.096	0.081	0.203	0.255	0.313	0.237	0.161	0.125	0.131	0.238	0.292	0.365
MACLR	✓	0.181	0.154	0.119	0.104	0.244	0.304	0.373	0.163	0.135	0.108	0.097	0.204	0.254	0.321
RTS	✓	0.187	0.153	0.120	0.106	0.242	0.304	0.382	0.186	0.151	0.121	0.108	0.227	0.283	0.354
free generation	✗	0.171	0.110	0.084	0.097	0.177	0.219	0.274	0.246	0.156	0.116	0.133	0.228	0.271	0.327
ICXML - content	✗	0.225	0.148	0.109	**0.141**	0.266	0.320	0.349	0.241	0.150	0.109	0.128	0.201	0.242	0.301
ICXML - label	✗	**0.234**	**0.160**	**0.119**	0.134	0.250	0.300	0.357	**0.278**	**0.169**	**0.125**	0.149	**0.246**	**0.290**	0.356
ICXML - retrieval	✓	0.220	0.155	0.115	0.135	**0.279**	**0.342**	**0.404**	0.252	0.141	0.105	**0.152**	0.225	0.256	**0.361**

Table 2: Experimental results obtained by the proposed approach and the baselines. The highest number in each column is bold-faced. For proposed approach, we are using GPT-3.5 for in-context learning.

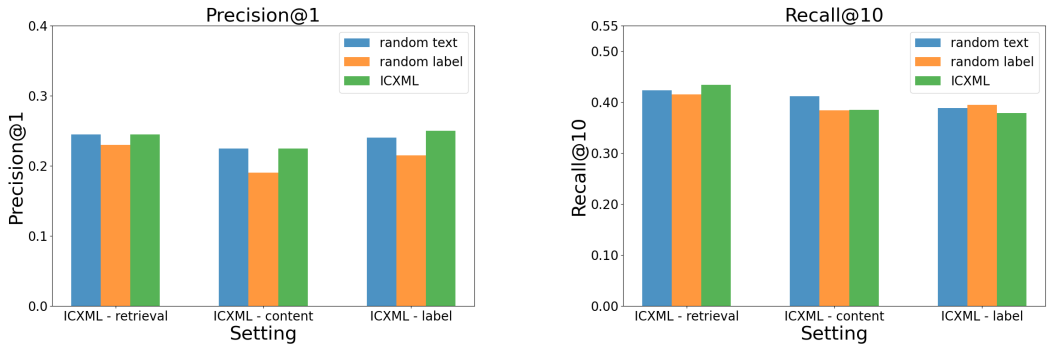

Figure 2: Results of different demonstration construction strategies on 200 samples from LF-Amazon-131K.

proach and the content-based approach comparably equivalent, whereas for LF-WikiSeeAlso-320K, the label-centric approach demonstrates a clear superiority. This discrepancy can be ascribed to the differing degrees of correlation between labels and primary identifiers (product names for LF-Amazon-131K and wiki page titles for LF-WikiSeeAlso-320K). The Amazon dataset demonstrates a stronger association between labels and product names, while in the case of the WikiSeeAlso, the labels appear to have a more substantial dependence on the actual content. Under our experimental framework where only product names or wiki page titles are generated, the relevance of each title-adapted pseudo label is comparatively low. Interestingly, even when provided with access to the input corpus and the ability to construct demonstrations through the retrieval of existing input sources, the performance of demonstration retrieval does not surpass that of demonstration generation. It exhibits better performance in terms of recall@10, but precision@1, intriguingly, is even superior in the case of generation. The reason could be diversity discrepancy between generated and retrieved inputs and noise in pairing fixed inputs with fixed label space. To gain deeper insights into these observed performance variations, we conducted an extensive analysis in Section 6.5.

6.5 Ablation Study

In this section, we study the top 1 and top 10 performance differences. Also, we conduct comprehensive ablation analyses to discern the contribution of each component in ICXML. These evaluations are performed on a sample comprising 200 instances from both test dataset. We use label-centroid generation. In our ablation study, we answer the following empirical research questions:

RQ1: *How is the effect of different demonstration construction strategies?*

We evaluate three distinct demonstration construction strategies on a subset of 200 instances extracted from LF-Amazon-131K test dataset. The strategies employed were as follows: retaining the original demonstrations, replacing the input text with random words, and replacing the paired labels with random labels. As presented in Table 2, it becomes evident that the most substantial decrease in Precision@1 performance occurs when the paired labels are replaced. This observation underscores the critical role played by label space coverage in

	LF-Amazon-131K							LF-WikiSeeAlso-320K						
generation and shortlisting	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
free generation	0.160	0.128	0.087	0.099	0.235	0.257	0.324	0.250	**0.174**	**0.136**	0.124	0.244	0.304	0.346
TAS-B	0.190	0.131	0.094	0.115	0.236	0.290	0.354	0.270	0.172	**0.136**	0.144	**0.262**	**0.314**	**0.381**
free + TAS-B results as hint	0.175	0.117	0.084	0.105	0.200	0.234	0.293	0.225	0.145	0.106	0.118	0.210	0.233	0.322
ICL	**0.220**	**0.140**	**0.107**	**0.135**	**0.252**	**0.312**	**0.361**	**0.310**	0.173	0.124	**0.164**	0.250	0.276	0.353

Table 3: Ablation study of generate stage on sampled LF-Amazon-131K and LF-WikiSeeAlso-320K datasets of size 200. "free generation" is the original in-context learning configuration with a reranking stage, "TAS-B" is applying reranking stage on top 100 results of TAS-B, "free + TAS-B results as hint" is using these top 100 labels as hint in free generation prompt. "ICL" is our proposed method. Reranking techniques are all based on LLM (GPT-3.5)

	LF-Amazon-131K							LF-WikiSeeAlso-320K						
reranking	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
heuristic	0.160	0.128	0.087	0.099	0.236	0.257	0.324	0.285	0.168	**0.127**	0.140	0.239	**0.283**	0.328
monoT5	0.180	0.117	0.089	0.119	0.215	0.272	0.345	0.220	0.137	0.105	0.111	0.190	0.229	0.281
LLM	**0.220**	**0.140**	**0.107**	**0.135**	**0.252**	**0.312**	**0.361**	**0.310**	**0.173**	0.124	**0.164**	**0.250**	0.276	**0.353**

Table 4: Ablation study of rerank stage on sampled LF-Amazon-131K and LF-WikiSeeAlso-320K datasets of size 200. "heuristic" can be regarded as natural results without reranking. "monoT5" is a pretrained ranking model. "LLM" is listwise reranking based on LLM.

the preparation of demonstrations. The results further imply the flexibility and efficacy of ICXML in handling XMC challenge with or without an input corpus.

RQ2: *How is the effect of generate and shortlisting?*

To discuss the effect of generate and shortlisting component, we simply modify the free generation setup, transitioning from an approximation of generated raw labels to our unique generate-rerank paradigm, which allows for an expanded scope of label mappings. Under this scheme, we regard the enlarged set via label space mapping of raw labels as a condensed candidate list, from which the top 10 are selected via listwise reranking based on GPT-3.5. In Table 3, "free generation" and "TAS-B" denotes further reranking from the top 100 results derived from the no-demonstration prompting configuration or TAS-B respectively. In an effort to rigorously evaluate the quality of the generated demonstration, we conducted an additional experimentation aiming at understanding the performance influence of input text that is exclusively generated by the LLM and moves beyond the boundaries of the accessible corpus. For this test, we used a prompt that was filled with pseudo labels serving as hints but did not include any pseudo pairs. Results are denoted as "free + TAS-B results as hint".

Comparing the results of all experiments, it is evident that our approach exhibits superior performance, affirming the effectiveness of demonstration generation. By feeding all soft matching results to LLM as hint to generate label candidate shortlist, "free + TAS-B results as hint" underperforms compared to the direct utilization of these results as candidates, highlighting the key role of external knowledge and inherent correlation between input and label conveyed by generated demonstrations.

RQ3: *How is the effect of label reranking?*

For this research question, we keep generated candidates frozen, but apply different strategies to produce the final top 10 answers. The strategies include: **Heuristic**: This strategy is incorporated within free generation configuration to opt for results from the generated label list that are most proximate to the mapped label space. For the ith instance, represent mapped labels based on generated labels Y_i as $\{y^1_{i,1}, \cdots, y^{10}_{i,1}, \cdots y^{10}_{i,k}\}$, where $\{y^1_{i,1}, \cdots, y^{10}_{i,1}\}$ are top 10 neighbor labels identified by an zero-shot retriever. Simply rerank Y_i with a heuristic rule: Specifically, the ranking is carried out in the order of the generated labels, where the nearest neighbor within the label candidate set is chosen. If this nearest neighbor has already been arranged, defer to the second nearest one. This procedure continues in a similar manner for subsequent generated labels. The final result should be $\{y^1_{i,1}, y^1_{i,2}, \cdots, y^{10}_{i,10}, \cdots y^{10}_{i,k}\}$. **MonoT5**: a ranking model which fundamentally leverages T5 architecture to calculate the relevance score, and is fine-tuned on MS MARCO passage dataset. Within the context of this comparison, we employ the monoT5-3B model variant for our analyses and evaluations. **LLM**: Our strategy introduced in sec-

		LF-Amazon-131K							LF-WikiSeeAlso-320K						
generate	reranking	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
GPT-3.5	GPT-3.5	0.220	0.140	0.107	0.135	0.252	0.312	0.361	**0.310**	0.173	0.124	**0.164**	0.250	0.276	0.353
GPT-3.5	GPT-4	0.220	**0.142**	0.105	0.135	0.243	0.310	**0.397**	0.295	**0.202**	**0.148**	0.148	**0.282**	**0.332**	**0.427**
Llama 2	RankVicuna	**0.225**	0.141	**0.108**	**0.140**	**0.256**	**0.316**	0.365	0.278	0.143	0.114	0.157	0.214	0.274	0.355

Table 5: Ablation study of different language models on sampled LF-Amazon-131K and LF-WikiSeeAlso-320K datasets of size 200. RankVicuna is fine-tuned on Llama 2 for zeroshot listwise reranking.

tion 4.3.

Experimental results of the three strategies are presented in Table 4. The observed enhancement when transitioning from heuristic to other strategies implies that increasing the number of neighbors incorporated within the label space mapping step can result in more correct labels being added to the shortlist. This suggests that a robust reranking approach has potential utility and validates the effectiveness of our generate-rerank framework. By broadening the candidate set, our framework provides an enriched space for accurate label selection, demonstrating its value in complex label space adaptations. Further improvement from monoT5 to LLM indicate the LLM's strong ability of multi-choice selection and reranking.

RQ4: *To what extent does the performance of our method generalize to different models?*

GPT-4 is confirmed to be significantly superior in terms of ranking performance (Sun et al., 2023). Due to the extensive size of the test sets, we confined our LLM-based listwise reranking using GPT-4 to a small subset of the dataset. The findings demonstrate that GPT-4 excels in reranking tasks, particularly with the WikiSeeAlso dataset. Here, our method, ICXML, consistently surpasses the baselines. These results underscore the potential capability of LLMs to adapt to extensive label spaces, thereby illustrating their utility in extreme classification scenarios.

Furthermore, to address the concerns raised about the reproducibility and robustness of our framework, particularly regarding the use of large black box models such as GPT 3.5/4.0, we have expanded our research to include experiments with the more recent and open-sourced large language model, Llama 2 (Touvron et al., 2023). In the generate and shortlisting stage, we used vanilla Llama 2 to construct demonstrations and generate candidate shortlist. In the reranking stage, we used RankVicuna (Pradeep et al., 2023), a model that has undergone instruction tuning and further distillation of knowledge derived from GPT-4's listwise reranking outcomes based on Llama 2. This was implemented to replicate our initial zeroshot listwise setup. The results, as shown in the Table 5, offer insights into the model-agnostic nature of our method and its efficacy across different large language model platforms.

7 Conclusions and Future Work

In conclusion, this paper addressed the challenges of Extreme Multi-Label Classification (XMC) in real-world scenarios with limited supervision signals. We proposed the ICXML framework to handle this setting without reliance on input text and pretraining. Experimental results demonstrated the effectiveness of our approach in improving the performance of XMC and its various zero-shot settings. Our research contributes to the advancement of XMC by offering new insights and methodologies for addressing real-world challenges.

For future work, an interesting direction would be to evaluate the adaptability of ICXML across diverse domains and multi-modal data. Understanding how the model behaves with different domain-specific terminologies and when combined with visual or auditory data will be crucial. Also, Combining ICXML with other state-of-the-art XMC techniques might offer synergistic benefits. Exploring hybrid models can potentially unlock new efficiencies and improved performance.

Risks and Limitations

Like other LLM based works, one of the risks of this work is ethical and bias considerations: Any biases present in the training data of ChatGPT will influence the generated labels. Without appropriate checks, these biases might amplify or result in misleading labels, especially in sensitive areas.

Furthermore, the adaptability of ICXML to data from diverse domains and in multi-modal formats is an area yet to be explored thoroughly. The behavior of the language model may vary based on the distinctiveness and intricacies of specific domain terminology or when integrating visual cues. Ad-

dressing these variations will be crucial for ICXML to be universally applicable.

References

Pranjal Aggarwal, Ameet Deshpande, and Karthik Narasimhan. 2023. Semsup-xc: Semantic supervision for zero and few-shot extreme classification. *arXiv preprint arXiv:2301.11309*.

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

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.

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

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.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2019. Large-scale multi-label text classification on eu legislation. *arXiv preprint arXiv:1906.02192*.

Wei-Lin Chen, Cheng-Kuang Wu, and Hsin-Hsi Chen. 2023. Self-icl: Zero-shot in-context learning with self-generated demonstrations. *arXiv preprint arXiv:2305.15035*.

Xiang Chen, Lei Li, Ningyu Zhang, Xiaozhuan Liang, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Decoupling knowledge from memorization: Retrieval-augmented prompt learning. *arXiv preprint arXiv:2205.14704*.

Daixuan Cheng, Shaohan Huang, Junyu Bi, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Furu Wei, Denvy Deng, and Qi Zhang. 2023. Uprise: Universal prompt retrieval for improving zero-shot evaluation. *arXiv preprint arXiv:2303.08518*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Kunal Dahiya, Nilesh Gupta, Deepak Saini, Akshay Soni, Yajun Wang, Kushal Dave, Jian Jiao, Prasenjit Dey, Amit Singh, Deepesh Hada, 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.

Kunal Dahiya, Deepak Saini, Anshul Mittal, Ankush Shaw, Kushal Dave, Akshay Soni, Himanshu Jain, Sumeet Agarwal, and Manik Varma. 2021. Deepxml: A deep extreme multi-label learning framework applied to short text documents. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 31–39.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*.

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.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122.

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*.

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.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

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.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao. 2023. Dr. icl: Demonstration-retrieved in-context learning. *arXiv preprint arXiv:2305.14128*.

Xinxi Lyu, Sewon Min, Iz Beltagy, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. Z-icl: Zero-shot in-context learning with pseudo-demonstrations. *arXiv preprint arXiv:2212.09865*.

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*.

A. Mittal, K. Dahiya, S. Malani, J. Ramaswamy, S. Kuruvilla, J. Ajmera, K. Chang, S. Agrawal, P. Kar, and M. Varma. 2022. Multimodal extreme classification. In *CVPR*.

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

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. *choice*, 2640:660.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. Rankvicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv preprint arXiv:2309.15088*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot reasoning. *arXiv preprint arXiv:2202.07206*.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7.

Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. 1995. Okapi at trec-3. In *Proceedings of the Third Text REtrieval Conference*, TREC-3, pages 109–126. Gaithersburg, MD: NIST.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523.

Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Tianyi Zhou, and Daxin Jiang. 2023. Large language models are strong zero-shot retriever. *arXiv preprint arXiv:2304.14233*.

Daniel Simig, Fabio Petroni, Pouya Yanki, Kashyap Popat, Christina Du, Sebastian Riedel, and Majid Yazdani. 2022. Open vocabulary extreme classification using generative models. *arXiv preprint arXiv:2205.05812*.

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*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Yuanhao Xiong, Wei-Cheng Chang, Cho-Jui Hsieh, Hsiang-Fu Yu, and Inderjit Dhillon. 2021. Extreme zero-shot learning for extreme text classification. *arXiv preprint arXiv:2112.08652*.

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

Yu Zhang, Zhihong Shen, Chieh-Han Wu, Boya Xie, Junheng Hao, Ye-Yi Wang, Kuansan Wang, and Jiawei Han. 2022b. Metadata-induced contrastive learning for zero-shot multi-label text classification. In *Proceedings of the ACM Web Conference 2022*, pages 3162–3173.

A Appendix: Instructions

See instructions on the next page.

t_1: generate demonstration input based on test input

Product title: `test input title`
Task: Please predict at least 5 relevant and diverse Amazon products titles.
Format: ["title1", "title2", "title3", "title4", "title5"], do not say any word or explain.
Product Description: `test input description`

Wiki title: `test input title`
Task: Please generate at least 5 relevant and diverse Wikipedia page titles.
Format: ["title1", "title2", "title3", "title4", "title5"], do not say any word or explain.
Wiki content: `test input title`

t_2: generate demonstration input based on label input

For an Amazon product recommendation task,
Product title: `test input title`
Candidate labels: `retrieved labels`
Task: For each label, guess an input title.
Format: ["title1", "title2", "title3", "title4", "title5"], each title is a guess based on a candidate label, title1 is a guess for first label, and so on. Only output one list and the list should be of size 30. do not explain or say anthing.

As 'See Also' pages of `test input title`
There's a list of Wikipedia page titles: `retrieved labels`
Task: For each page, generate a "See also" page title.
Format: ["title1", "title2", "title3", "title4", "title5"], each title is a guess based on a candidate label, title1 is a guess for first label, and so on. Only output one list and the list should be of size 30. do not explain or say anthing.

t_3: Inference

Product title: `demonstration input title`
Relevant product: `corresponding labels`
... ...
Task: Please predict at least 10 relevant products for a new Amazon product title: `test input title`
Product Description: `test input description`
Format: Only output titles with line break, do not include anything else.

Wiki title: `demonstration input title`
'See Also' pages: `corresponding labels`
... ...
Title: `test input title`
Content: `test input description`
Task: Generate 'See also' suggestions related to the Wikipedia title `test input title`
Format: Only output titles with line break, do not include anything else.

t_4: Reranking

Task:	Given a query product, select the top 10 most relevant products from a list of candidates.
Query product title:	`test input title`
Format:	A list of integers representing the indices of the top 10 most possible titles. Example: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Candidates:	`label shortlist`
Product Description:	`test input description`

Task:	From the following candidate list of Wikipedia pages, select top 10 that would be most relevant for the 'See also' section of the given page:
Wiki title:	`test input title`
Format:	A list of integers representing the indices of the top 10 most possible titles. Example: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Candidates:	`label shortlist`
Wiki Content:	`test input description`