Extreme Zero-Shot Learning for Extreme Text Classification

Yuanhao Xiong*
UCLA
yhxiong@cs.ucla.edu

Wei-Cheng Chang
Amazon Inc.
weicheng.cmu@gmail.com

Cho-Jui Hsieh
UCLA
chohsieh@cs.ucla.edu

Hsiang-Fu Yu
Amazon Inc.
rofu.yu@gmail.com

Inderjit Dhillon
UT Austin
inderjit@cs.utexas.edu

Abstract

The eXtreme Multi-label text Classification (XMC) problem concerns finding most relevant labels for an input text instance from a large label set. However, the XMC setup faces two challenges: (1) it is not generalizable to predict unseen labels in dynamic environments, and (2) it requires a large amount of supervised (instance, label) pairs, which can be difficult to obtain for emerging domains. In this paper, we consider a more practical scenario called Extreme Zero-Shot XMC (EZ-XMC), in which no supervision is needed and merely raw text of instances and labels are accessible. Few-Shot XMC (FS-XMC), an extension to EZ-XMC with limited supervision is also investigated. To learn the semantic embeddings of instances and labels with raw text, we propose to pre-train Transformer-based encoders with self-supervised contrastive losses. Specifically, we develop a pre-training method **MACLR**, which thoroughly leverages the raw text with techniques including **M**ulti-scale **A**daptive **C**lustering, **L**abel **R**egularization, and self-training with pseudo positive pairs. Experimental results on four public EZ-XMC datasets demonstrate that MACLR achieves superior performance compared to all other leading baseline methods, in particular with approximately 5-10% improvement in precision and recall on average. Moreover, we show that our pre-trained encoder can be further improved on FS-XMC when there are a limited number of ground-truth positive pairs in training. Our code is available at https://github.com/amzn/pecos/tree/mainline/examples/MACLR.

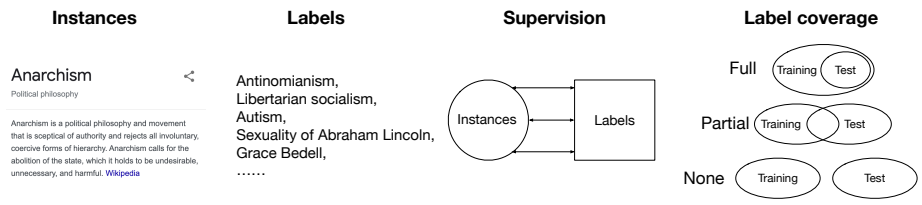
1 Introduction

The e**X**treme **M**ulti-label text **C**lassification (XMC) problem aims at tagging a text input with most relevant subset of labels from an extremely large output space. Many web-related applications can be formulated as an XMC task with encouraging results, such as finding the best matching products from a large catalog in e-commerce systems (Medini et al., 2019; Chang et al., 2021), auto-completing queries given its prefix on search engines (Yadav et al., 2021), predicting search keywords for dynamic advertising (Prabhu et al., 2018; Chang et al., 2020b), tagging categories of Wikipedia articles from a large label taxonomy (Dekel and Shamir, 2010; Chalkidis et al., 2019), to name just a few.

The current XMC setup is built on full label coverage and full supervision, where full label coverage means labels to be predicted have appeared in the training set and full supervision indicates it requires a significant number of annotated (instance, label) pairs. In detail, it is assumed that an XMC algorithm has access to raw text of instances and labels, together with their corresponding relations during training, as shown in Figure 1.

However, there are several limitations of this XMC setting. First of all, due to the assumption of full label coverage, it is typical in XMC approaches to simply treat labels as IDs for classification and thus they are restricted to making predictions within observed labels. This assumption is unrealistic since the label set usually keeps growing over time, e.g., newly added websites or products which are absent during training yet crucial for applications such as recommendation and advertising. Besides, collecting labeled pairs is time-consuming, expensive and sometimes infeasible, for example, launching an e-commerce system in the emerging locale, where no user behavioral signals are avaiable. In spite of these constraints, most existing methods (Dahiya et al., 2021b; You et al., 2019; Mittal et al., 2021; Dahiya et al., 2021a) followed this XMC setup. It can be seen in Figure 2 that Astec (Dahiya et al., 2021b), one of the state-of-the-art extreme classifiers, is incapable of handling the scenario without supervision, which leads to zero performance in both Precision@5 and Recall@100. Moreover, the increasing trend in Astec's perfor-

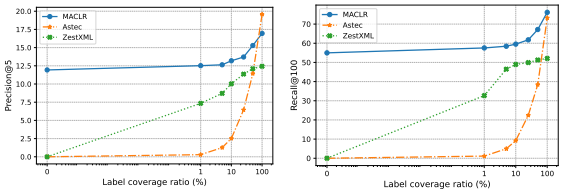
*Work done during an internship at Amazon Inc.



	Instances	Labels	Supervision	Label coverage
XMC	√	√	√	Full
GZ-XMC	√	√	√	Full & Partial & None
FS-XMC	√	√	√*	Full & Partial & None
EZ-XMC	√	√	×	Full & Partial & None

Figure 1: Four different settings in XMC. Four essential components are considered: instances (raw text), labels (raw text), supervision (positive pairs), and label coverage. In detail, we divide label coverage into 3 groups: full, partial, and none. * in FS-XMC emphasizes that only a limited amount of supervision is available. We can see that EZ-XMC is the most general and practical setting, where no supervision and label coverage is required.

mance along with the label ratio suggests that it depends highly on the supervision level and is hard to generalize to unseen labels. This motivates us to investigate how to design an effective XMC model with zero supervision.



(a) Precision@5. (b) Recall@100.

Figure 2: Performance of three representative XMC methods on LF-Amazon-131K at different ratios of label coverage. A subset covering $[0, 1, 5, 10, 25, 50, 100](\%)$ of the whole label set is sampled for fine-tuning.

In this paper, we consider an essential yet under-explored XMC setting, called Extreme Zero-shot XMC (EZ-XMC). As depicted in Figure 1, we can access raw text of both instances and labels but do not know their corresponding relations in EZ-XMC. Moreover, we do not make any assumption on the label coverage, so the labels in the testing set may or may not appear in the training stage. An extension to EZ-XMC with a limited number of training pairs, Few-shot XMC (FS-XMC), is also taken into account in our paper. Either EZ-XMC or FS-XMC occurs frequently in the real world since informative and abundant (instance, label) pairs are never easy to obtain. Also, it is more practical and worthwhile to reduce labor for manual annotation by solving problems under EZ-XMC. Note that generalized zero-shot XMC (GZ-XMC) proposed in a recent work (Gupta et al., 2021) can be regarded as a special case of EZ-XMC. GZ-XMC allows that the set of test labels is not completely overlapped with training labels but still requires supervision from positive pairs, as shown in Figure 1. From Figure 2, we can observe that ZestXML (Gupta et al., 2021) designed for GZ-XMC also suffers the issue of no supervision.

A natural question then arises: how should we deal with EZ-XMC problems? Despite the name, EZ-XMC is barely easy to tackle. Fortunately, although dedicated supervision signals are lacking, raw text of instances and labels, e.g., product descriptions and categories, are still accessible in EZ-XMC. Thus it is of vital importance to effectively leverage self-information of these data to train a model for classification. To overcome challenges encountered in EZ-XMC, we turn to solving the problem from a different perspective without learning classifiers explicitly. In particular, XMC can be cast into a problem which learns a sentence encoder $\mathcal{E}$ to map instances and labels into dense embeddings, and predictions are made through approximate nearest neighbor search algorithms in the latent space (Shrivastava and Li, 2014). Motivated by recent progresses in self-supervised learning (Gao et al., 2021; Chen et al., 2020; He et al., 2020b; Devlin et al., 2019), we propose **MACLR** (**M**ulti-scale **A**daptive **C**lustering & **L**abel **R**egularization), a two-stage pre-training procedure with those unpaired raw data to obtain a sentence encoder $\mathcal{E}$ under EZ-XMC. As to FS-XMC, fine-tuning the encoder on a few paired data is sufficient for the performance boost. Figure 2 demonstrates that MACLR achieves superior performance when no supervision is available and achieves much higher recall than Astec and ZestXML by a large margin,

even under the higher label coverage ratio.

Our main contributions are summarized below:

- We propose an essential Extreme Zero-Shot XMC (EZ-XMC) setting without any assumptions on supervision and label coverage, which has not been explored in previous work and is more practical in real applications.
- We leverage unlabeled data to pretrain the sentence encoder $\mathcal{E}$ with improved Inverse Cloze Task in Stage I of MACLR. In particular, multi-scale adaptive clustering and label regularization are proposed to utilize raw text thoroughly. In Stage II, we further self-train the encoder with pseudo positive pairs constructed from $\mathcal{E}$ in Stage I as well as TF-IDF model with complementary information.
- Comprehensive experiments are conducted on four public benchmark EZ-XMC datasets. Results demonstrate that our pre-trained encoder can outperform existing unsupervised baseline methods notably. As an example, MACLR achieves Recall@100 of 54.99%, nearly the same level as Astec (one of the SOTA XMC methods) (Dahiya et al., 2021b) trained with a supervised subset covering around 70% labels on LF-Amazon-131K.
- MACLR can also achieve comparable or even better performance under the few-shot setting than those models heavily dependent on supervised information. For example, MACLR is better than the SOTA ZestXML (Gupta et al., 2021) in Recall@100 over 20% (57.55% v.s. 32.69%) when fine-tuned on the subset covering 1% labels of LF-Amazon-131K.

2 Related Work

Extreme multi-label classification Various extreme classifiers have been proposed to address the large output space challenge of XMC problems. We can broadly categorize them into two groups: partitioned-based models with linear classifiers (Prabhu et al., 2018; Prabhu and Varma, 2014; Yu et al., 2020) that partition labels with hierarchical trees, leading to sub-linear inference time complexity, and embedding-based methods (Bhatia et al., 2015; Jain et al., 2019; Guo et al., 2019) that learn a classifier for each label and leverage approximated nearest neighbor (Malkov and Yashunin, 2018; Guo et al., 2016) to index labels in the large output space. There are also deep learning models such as AttentionXML (You et al., 2019), Astec (Dahiya et al., 2021b), SiameseXML (Dahiya et al., 2021a), and XR-Transformer (Zhang et al., 2021) that further improve the accuracy of those linear counterparts with various advanced encoder architectures. Nevertheless, none of those XMC methods can handle the EZ-XMC setup: they not only suffer from the lack of supervised signals, but also fail to generalize to unseen cold-start labels in the test set. The only exception is ZestXML (Gupta et al., 2021), a recently proposed XMC method that was designed to address the generalized zero-shot XMC (GZ-XMC) problem where a number of labels for prediction are absent during training. While ZestXML partially resolves the generalization challenge of cold-start labels, just like those conventional XMC models, it still depends heavily on a large number of training data with positive (instance, label) pairs.

Self-supervised learning techniques The past few years have witnessed great promise in self-supervised learning (Lan et al., 2020; Chen et al., 2020; He et al., 2020b; Devlin et al., 2019; Khosla et al., 2020; Gao et al., 2021), where a pre-training task is defined using only data's self-information. Learned representations from the pre-training task can be then leveraged in a wide range of downstream tasks in various domains, such as image classification (Chen et al., 2020; He et al., 2020b) and object detection (Li et al., 2020) in computer vision, and open-domain question answering (Lee et al., 2019; Guu et al., 2020) in natural language processing. Specifically, we focus on contrastive approaches for Sentence-BERT (Reimers and Gurevych, 2019) models in this paper, where the intuition is to pull semantically close neighbors together and push apart non-neighbors via noise contrastive estimation or N-pair losses. Various effective pre-training tasks such as Inverse Cloze Task (ICT) (Lee et al., 2019) and SimCSE (Gao et al., 2021) have been shown to improve the performance of Sentence-BERT models.

3 Problem Formulation

In this section, we present the problem formulation of EZ-XMC. With $\mathcal{X}$ and $\mathcal{Y}$ denoting the set of instances and labels respectively, the general XMC problem can be viewed as learning a scoring function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. $f(\cdot, \cdot)$ maps an (instance, label) pair (x, y) to a similarity score, which is used to make a prediction through approximate nearest neighbor search algorithms. In previous set-

tings such as XMC and GZ-XMC, a considerable amount of relevant (instance, label) pairs $\{(x_i, y_i)\}$ are available. On the contrary, in EZ-XMC, we have no knowledge about corresponding relations between instances and labels, but only their raw text, as shown in Figure 1. In this case, existing approaches that depend on the relevant pairs fail to learn an effective scoring function, even with a few paired data under FS-XMC.

Recent progresses in self-supervised learning have shown that a generalized sentence encoder can be learned through elaborately designed pre-training tasks even without any supervision (Lee et al., 2019; Chang et al., 2020a), and then adapted to different downstream tasks directly or via slight finetuning. On the other hand, the scoring function f can be modeled as $f(x, y) = \langle \mathcal{E}(x), \mathcal{E}(y) \rangle$, where $\mathcal{E}$ is a sentence encoder producing semantical dense embeddings, and $\langle \cdot, \cdot \rangle$ is the similarity measurement such as inner product and cosine similarity. Without loss of generality, inner product is adopted in the paper as the similarity metric between embeddings of instances and labels. Thus, we formulate the problem as training an encoder $\mathcal{E}$ with raw text of $\mathcal{X}$ and $\mathcal{Y}$ through a pre-training task for EZ-XMC. As to the few-shot scenario FS-XMC, we can fine-tune $\mathcal{E}$ for improvement.

4 Method

In this section, we introduce a two-stage pre-training procedure, MACLR, to thoroughly leverage unpaired data with raw text for EZ-XMC. Specifically, we present the general framework in Section 4.1, and then dive into details of two stages, pre-training with the improved Inverse Cloze Task and self-training with pseudo positive pairs, in Sections 4.2 and 4.3 respectively. A complete algorithm is presented in Algorithm 1 in Appendix 4.4.

4.1 Framework

The framework of our pre-training procedure is shown in Figure 3. MACLR consists of two stages:

- Stage I: title-context pairs are constructed for the Inverse Cloze Task, and the encoder $\mathcal{E}$ is then trained on these pairs together with two proposed techniques, multi-scale adaptive clustering and label regularization.
- Stage II: More pseudo positive pairs are crafted using different score functions modeled by the encoder from Stage I and TF-IDF respectively. $\mathcal{E}$ is further trained on additional pairs to improve the encoding performance.

Details of each component in our pre-training framework are discussed in the following sections.

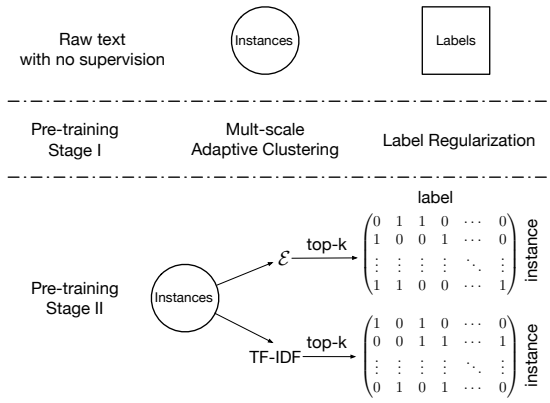


Figure 3: Framework of our pre-training procedure.

4.2 Stage I: Pre-training with improved ICT

Inverse Cloze Task (Lee et al., 2019) is a frequently used pre-training task for the sentence encoder. Specifically, for an instance $x = \{s_1, \ldots, s_n\}$ consisting of n sentences, ICT randomly samples a sentence to serve as the pseudo positive label $\hat{y} = s_i$ where $i \sim [1, n]$. Then the rest of x is the pseudo instance $\hat{x} = \{s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n\}$. In XMC, due to the property that the label usually summarizes the instance with one short sentence, which works similarly as the title s_1, we directly utilize (context, title) pairs in the form of $(\hat{x} = \{s_2, \ldots, s_n\}, \hat{y} = s_1)$. This construction works as the analog of the ground truth (instance, label) pairs and capture the semantics of a sentence. With these pseudo pairs, the contrastive training objective for a mini-batch of N pairs is as follows:

$$\mathcal{L}_{\text{contrastive}} = -\sum_{i=1}^{N} \log \frac{\exp(\mathcal{E}(\hat{x}_i) \cdot \mathcal{E}(\hat{y}_i))}{\sum_{j=1}^{N} \exp(\mathcal{E}(\hat{x}_i) \cdot \mathcal{E}(\hat{y}_j))} \quad (1)$$

Based on ICT, we also develop two techniques, multi-scale adaptive clustering and label reguarization, to fully leverage the information of unpaired instances and labels.

4.2.1 Multi-scale Adaptive Clustering

In the original ICT scheme, we can construct only one positive pair for a particular instance. It is relatively hard in contrastive learning without enough positive examples, especially for extreme multi-label classification where one instance might be associated with more than one label, and a label is also likely to point to several different instances at the same time. Thus a question arises naturally:

is it possible to construct more positive pairs from purely unpaired raw data to intergrate richer information into the pre-training process? We solve it by the unsupervised K-means clustering. In detail, we divide pseudo (context, title) pairs from ICT into K clusters through K-means based on the embeddings of all instances. Then if $C(\hat{x}_i) = C(\hat{x}_j)$, i.e., $\hat{x}_i$ and $\hat{x}_j$ belong to the same cluster, $(\hat{x}_i, \hat{y}_j)$ and $(\hat{x}_j, \hat{y}_i)$ are regarded as positive pairs besides original ICT pairs. Furthermore, supervised contrastive loss is adopted for training the encoder with a mini-batch of N pairs based on the cluster assignment:

$$\mathcal{L}_{\text{cluster}} = \sum_{i=1}^{N} \frac{-1}{|P_{\mathcal{Y}}(i)|} \sum_{p \in P_{\mathcal{Y}}(i)} \log \frac{\exp(\mathcal{E}(\hat{x}_i) \cdot \mathcal{E}(\hat{y}_p))}{\sum_{j=1}^{N} \exp(\mathcal{E}(\hat{x}_i) \cdot \mathcal{E}(\hat{y}_j))} \tag{2}$$

Here, $P_{\mathcal{Y}}(i) = \{p \in \{1, \ldots, N\} : C(\hat{x}_i) = C(\hat{x}_p)\}$ is the set of indices of all positives for $\hat{x}_i$ in the batch, and $|P_{\mathcal{Y}}(i)|$ is its cardinality. Minimizing Equation (2) pulls close the representations of instances and their positive labels within the same cluster and pushes away the representations of those from different clusters.

Besides, since the ultimate goal is the minimization of Equation (1), we propose a multi-scale approach with adaptive training, which guides the encoder to learn the easier tasks with sufficient positive examples, and then master harder tasks gradually. This approach allows the encoder to learn from the coarse scale to the fine scale of clustering assignment, and is similar to the idea of curriculum learning (Bengio et al., 2009) to first focus on learning from a subset of simple examples, and expanding to include the remaining harder samples. Our adaptive training process can be conducted by modifying the cluster size to adjust the task difficulty accordingly. To be specific, we initialize the cluster assignment with the number of clusters K_0, and double the cluster size every T steps. The cluster assignment is also updated every T_{update} steps along with the training of $\mathcal{E}$. Such a process lasts for half of the total training steps T_{total} to take advantage of positive examples from constructed clusters. The obtained intermediate encoder from this adaptive procedure is expected to satisfactorily capture the semantics of a sentence and is ready to deal with the optimization of Equation (1). Then for the rest half of training steps, we turn to the hardest setting treating each instance as one independent cluster, which exactly falls into the contrastive training objective in Equation (1). Our multi-scale adaptive clustering is illustrated in Figure 4.

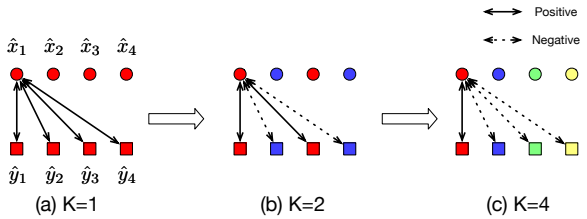


Figure 4: An example of multi-scale adaptive clustering. Here different colors represent different clusters. (a) In the beginning, there is only one cluster and $\{\hat{y}_j\}_{j=1}^{4}$ are all positive labels for $\hat{x}_1$. (b) K is doubled to 2 and now $\hat{y}_1$ and $\hat{y}_3$ are positive to $\hat{x}_1$. (c) Finally, K is equal to 4 where each instance itself is a cluster, and hence $\hat{x}_1$ only has one positive label $\hat{y}_1$. The process is similar for the rest of the instances.

4.2.2 Label Regularization

In addition to leveraging information from the instance side, we also have access to the raw texts of the whole label set and can utilize them to boost the encoder's performance from the label side (Mueller et al., 2022; Müller et al., 2022). Intuitively, for a randomly sampled label, with a high probability it is an negative example to the instance of interest. We can take advantage of this intuition to make the embedding of the instance far from its irrelevant labels. Instead of increasing the distance directly, it is more stable and effective to adopt contrastive losses. To avoid overfitting, we choose a new positive example for each instance instead of its corresponding pseudo label from ICT which has been used in $\mathcal{L}_{\text{cluster}}$. More concretely, $\hat{x}_i^+$ is selected exactly the same as $\hat{x}_i$, since the dropout layer is placed in the standard training of Transformer-based models and can be viewed as a minimal form of data augmentation (Gao et al., 2021). By feeding the same sentence to the encoder $\mathcal{E}$, two embeddings with different dropout masks are obtained, i.e., $\hat{h}_i = \mathcal{E}(\hat{x}_i, z_i)$ and $\hat{h}_i^+ = \mathcal{E}(\hat{x}_i^+, z_i^+)$ where z represents a random mask for dropout. $\hat{h}_i \neq \hat{h}_i^+$ due to the dropout noise, but they hold similar semantics from the same sentence and thus can be used as a positive pair for contrastive learning. The procedure of label regularization is depicted in Figure 5. At each step, we sample M real labels from the label set $\mathcal{Y}$, and the reguarization term is computed as follows:

$$\mathcal{L}_{\text{label}} = \sum_{i=1}^{N} -\log \frac{\exp(\hat{h}_i \cdot \hat{h}_i^+)}{\sum_{j=1}^{M} \exp(\hat{h}_i \cdot \mathcal{E}(y_j^-)) + \exp(\hat{h}_i \cdot \hat{h}_i^+)} \tag{3}$$

Through minimizing $\mathcal{L}_{\text{label}}$, the encoder learns to pull the instance away from its irrelevant labels

and incorporate the dropout augmentation at the same time. Together with $\mathcal{L}_{\text{cluster}}$, we have the final objective function for pre-training in the Stage I as

$$\mathcal{L} = \mathcal{L}_{\text{cluster}} + \mathcal{L}_{\text{label}}. \tag{4}$$

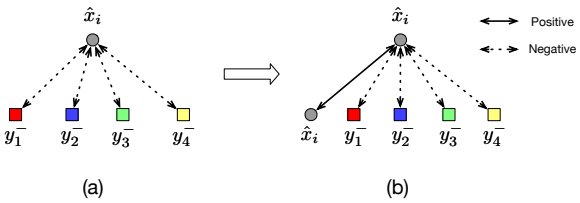


Figure 5: An illustration of label reguarization. (a) shows that $\hat{x}_i$ is expected to be far from sampled irrelevant labels $\{y_j^-\}_{j=1}^4$, while (b) indicates the identical $\hat{x}_i$ is added as a positive example for label regularization.

4.3 Stage II: Self-training with multi-viewed pseudo pairs

After the pre-training procedure in Section 4.2, we can obtain an intermediate encoder $\mathcal{E}_I$. But are there any ways to further improve the encoder? Inspired by self-training in semi-supervised learning (Yalniz et al., 2019; Xie et al., 2020; He et al., 2020a; Zoph et al., 2020), $\mathcal{E}_I$ can be leveraged to make predictions on those unpaired training instances themselves, to generate pseudo positive pairs. These pseudo pairs are much better than random guessing and can serve as a distinct view from ICT pairs. On the other hand, similar pseudo pairs can be constructed by other unsupervised methods such as TF-IDF, which provide different and complementary information about the instance.

With multi-viewed pseudo positive pairs, we can conduct further training on the encoder in State II from a new perspective and self-improve $\mathcal{E}_I$. The detailed process works as follows:

1) Compute the similarity score using $\mathcal{E}_I$ for each training instance x_i, and select labels with top-k maximum scores as its pseudo labels;
2) Generate labels similarly with TF-IDF, except that $\mathcal{E}(x)$ and $\mathcal{E}(y)$ are replaced with their TF-IDF vectors;
3) Mix pseudo positive pairs from 1) and 2) together, and train $\mathcal{E}_I$ on them with Equation (2).

4.4 MACLR Algorithm

The whole pre-training procedure of MACLR is shown in Algorithm 1. Note that for FS-XMC, we simply fine-tune the encoder $\mathcal{E}$ from MACLR on available positive pairs for several steps by minimizing the original contrastive loss in Equation (1).

Algorithm 1 Pre-training procedure of MACLR

Input: Raw text of instances and labels $(\mathcal{X}, \mathcal{Y})$, the sentence encoder $\mathcal{E}$, batch size N and M, training step parameters T_K, T_{update} and T_{total}, initial cluster size K_0, # of top candidates k

Output: A pre-trained sentence encoder $\mathcal{E}$

▷ Stage I: Pre-training with the improved ICT

Construct ICT (context, title) pairs from raw texts in $\mathcal{X}$

Feed the context for each pair into the encoder $\mathcal{E}$ and cluster them into $K = K_0$ clusters via k-means

for $t = 1, \ldots, T_{\text{total}}$ **do**

  Sample a mini-batch of pseudo pairs of size N and a mini-batch of real labels of size M

  Compute the loss: $\mathcal{L} = \mathcal{L}_{\text{cluster}} + \mathcal{L}_{\text{label}}$

  Train the encoder by minimizing $\mathcal{L}$

  if $t \bmod T_K = 0$ and $t < T_{\text{total}}/2$ **then**

    $K = K * 2$

  end if

  if $t \bmod T_{\text{update}} = 0$ and $t < T_{\text{total}}/2$ **then**

    Feed raw texts of $\mathcal{X}$ again into $\mathcal{E}$, and update current cluster assignment via k-means with the cluster number K

  end if

  if $t \geq T_{\text{total}}/2$ **then**

    Treat each instance as an independent cluster

  end if

end for

▷ Stage II: Self-training with multi-viewed pseudo pairs

Construct pseudo pairs $(\mathcal{X}_{\text{pseu}}, \mathcal{Y}_{\text{pseu}})$ by selecting top-k candidate labels with the similarity metric on the encoder $\mathcal{E}$ and TF-IDF respectively

Train the encoder $\mathcal{E}$ for T_{total} steps by minimizing Equation (2)

5 Experimental Results

5.1 Experimental Settings

Datasets We evaluate our proposed MACLR on 4 public XMC benchmark datasets (Bhatia et al., 2016; Gupta et al., 2021) where raw text of instances and labels are available. These datasets are derived from real-world applications, ranging from item-to-item recommendation (LF-Amazon-131K, LF-Amazon-1M), to Wikipedia articles category/title tagging (LF-WikiSeeAlso-320K, LF-

Wikipedia-500K). Detailed dataset statistics are presented in Table 5 in Appendix A.

Evaluation Protocol We consider two evaluation setups: Extreme Zero-shot Learning of XMC (EZ-XMC) and Few-shot Learning of XMC (FS-XMC). EZ-XMC is a fully unsupervised learning setup where no positive (instance, label) pairs are available. The only available information is the raw text of training instances and the whole label set. FS-XMC is a semi-supervised learning setup where only very few positive (instance, label) pairs in the training set are available. Regardless of the learning procedure, all models are evaluated on the same test set for fair comparison.

We evaluate the models' performance with precision@k (P@k, $k \in \{1, 3, 5\}$) and recall@k (R@k, $k \in \{1, 3, 5, 10, 100\}$), which are two commonly-used evaluation metrics in the XMC literature (Reddi et al., 2019; Chang et al., 2021).

Baseline Methods For EZ-XMC, we compare our method with the following unsupervised learning algorithms: TF-IDF, XR-Linear, GloVe, SentBERT, MPNet, SimCSE and ICT. Note that SentBERT and MPNet are pre-trained on external multi-task learning datasets with extra supervision. In contrast, SimCSE and ICT are fully unsupervised pre-rained Siamese-Transformers on the specific XMC dataset only. Detailed description of each method can be found in Appendix A.

For FS-XMC, as few-shot (instance, label) pairs are available, we additionally compare fine-tuned MACLR with competitive XMC approaches, including Astec (Dahiya et al., 2021b), SiameseXML (Dahiya et al., 2021a), and ZestXML (Gupta et al., 2021). ZestXML is the leading XMC method that improves performance on few-shot labels. We also take into account SentBERT (Reimers and Gurevych, 2019) with further fine-tuning to demonstrate the effectiveness of our pre-training procedure.

5.2 Zero-Shot Learning

In this section, we focus on extreme zero-shot learning (EZ-XMC), where no real positive (instance, label) pairs are accessible. Table 1 presents detailed performance of precision and recall on all four datasets. Our proposed MACLR consistently outperforms all comparing baselines by a large margin on all four datasets. Compared to the leading sparse method TF-IDF, MACLR has an average of 5.3% and 9.1% absolute improvement in Precision@1 and Recall@100, respectively. Compared to the leading neural model MPNet, MACLR has an average of 3.5% and 10.9% absolute improvement in Precision@1 and Recall@100, respectively.

Table 1: Extreme Zero-shot Learning (EZ-XMC) comparison of different unsupervised methods.

Method	Precision @1	Precision @3	Precision @5	Recall @1	Recall @3	Recall @5	Recall @10	Recall @100
LF-Amazon-131K								
TF-IDF	12.38	11.50	9.14	6.91	18.14	23.21	29.32	45.04
XR-Linear	7.56	7.84	7.30	4.05	12.11	18.32	29.17	40.39
GloVe	3.67	2.78	2.15	2.05	4.33	5.44	7.23	14.17
SentBERT	1.86	1.44	1.14	1.01	2.22	2.88	4.01	10.18
MPNet	13.94	11.41	8.82	7.82	18.08	22.58	27.91	43.39
SimCSE	10.13	8.61	6.69	5.61	13.39	16.84	21.27	35.81
ICT	13.82	11.41	8.90	7.76	18.09	22.80	28.94	47.40
MACLR (ours)	**18.13**	**15.42**	**11.93**	**10.35**	**24.45**	**30.43**	**37.28**	**54.99**
LF-WikiSeeAlso-320K								
TF-IDF	10.71	8.90	7.15	5.92	13.03	16.48	21.60	42.55
XR-Linear	4.73	4.27	3.90	2.23	5.83	8.64	14.18	36.93
GloVe	3.86	2.76	2.21	2.12	4.11	5.22	6.95	15.33
SentBERT	1.71	1.27	1.06	1.08	2.16	2.90	4.17	10.76
MPNet	13.75	11.93	9.58	8.14	17.77	22.21	28.11	45.91
SimCSE	9.03	6.64	5.22	4.99	9.89	12.34	15.93	30.11
ICT	10.76	10.05	8.12	6.12	14.32	18.05	23.01	39.77
MACLR (ours)	**16.31**	**13.53**	**10.78**	**9.71**	**20.39**	**25.37**	**32.05**	**53.83**
LF-Wikipedia-500K								
TF-IDF	20.30	12.98	9.96	7.25	12.91	15.98	20.31	38.16
XR-Linear	10.67	8.77	7.61	3.69	8.58	12.11	19.80	31.02
GloVe	2.19	1.52	1.23	0.85	1.66	2.18	3.10	8.52
SentBERT	0.17	0.15	0.13	0.05	0.13	0.18	0.30	1.29
MPNet	22.46	12.87	9.49	8.74	14.07	16.76	20.64	34.72
SimCSE	14.32	6.84	4.55	4.24	8.03	11.26	14.35	27.68
ICT	17.74	9.67	7.06	7.35	11.60	13.84	17.19	31.08
MACLR (ours)	**28.44**	**17.75**	**13.53**	**10.40**	**18.16**	**22.38**	**28.52**	**50.09**
LF-Amazon-1M								
TF-IDF	7.68	9.20	7.23	5.61	19.30	24.92	31.76	51.79
XR-Linear	5.19	5.48	5.26	3.63	11.30	17.94	31.18	43.79
GloVe	4.05	4.07	3.07	2.91	8.42	10.44	12.90	21.18
SentBERT	2.82	2.87	2.13	2.03	5.91	7.21	8.80	14.22
MPNet	8.29	8.87	6.80	6.04	18.64	23.51	29.35	46.15
SimCSE	3.33	3.69	2.74	2.38	7.66	9.38	11.43	18.54
ICT	8.66	9.26	7.13	6.30	19.45	24.60	30.73	48.42
MACLR (ours)	**9.58**	**10.41**	**8.03**	**7.38**	**22.01**	**27.72**	**34.48**	**55.23**

Speaking of sparse lexical matching approaches, TF-IDF remains a tough-to-beat unsupervised baseline. Specifically, TF-IDF performs better than many BERT variants (e.g., SentBERT, SimCSE, ICT), which is aligned with the finding in recent zero-shot dense retrieval literature (Thakur et al., 2021; Izacard et al., 2022). It suggests the importance of designing proper self-supervised learning tasks for Transformer models in unsupervised EZ-XMC setup. Note that XR-Linear is based on TF-IDF vectors whereas the noise from pseudo pairs makes it even inferior to the original TF-IDF.

As for pre-trained SentBERT models, on the other hand, only MPNet shows comparable performance with TF-IDF. MPNet remains competitive because it was trained on a large supervised corpus (out-of-domain) to learn semantics between paraphrasing sentences. Thus, MPNet should be viewed as a multi-task learning baseline with ex-

Table 2: Results of FS-XMC where the training subset covers 1% labels from the whole set.

Method	Precision			Recall				
	@1	@3	@5	@1	@3	@5	@10	@100
LF-Amazon-131K								
XR-Linear	1.53	0.57	0.36	0.67	0.75	0.78	0.81	0.92
Astec	0.94	0.44	0.29	0.55	0.78	0.84	0.91	1.13
SiameseXML	1.45	0.56	0.35	0.84	0.96	1.00	1.03	1.16
ZestXML	10.10	9.19	7.34	5.63	14.46	18.61	23.73	32.69
SentBERT	12.64	9.82	7.80	6.97	15.34	19.74	25.33	43.53
MPNet	14.78	11.55	8.97	8.28	18.24	22.84	28.54	45.89
MACLR (ours)	**18.74**	**16.07**	**12.52**	**10.73**	**25.44**	**31.89**	**39.17**	**57.55**
LF-WikiSeeAlso-320K								
XR-Linear	1.24	0.57	0.37	0.42	0.58	0.63	0.68	0.76
Astec	1.25	0.60	0.41	0.69	0.98	1.11	1.27	1.56
SiameseXML	1.81	0.75	0.48	1.03	1.26	1.33	1.41	1.67
ZestXML	8.74	6.78	5.41	4.68	9.70	12.21	15.73	24.98
SentBERT	16.30	12.62	10.08	9.30	18.92	23.78	30.40	52.92
MPNet	17.14	12.64	9.96	9.97	18.98	23.45	29.67	50.75
MACLR (ours)	**19.09**	**14.57**	**11.53**	**11.39**	**22.34**	**27.63**	**34.81**	**57.92**
LF-Wikipedia-500K								
XR-Linear	2.95	1.19	0.75	0.62	0.74	0.76	0.79	0.84
Astec	2.85	1.16	0.73	1.46	1.75	1.84	1.92	2.08
SiameseXML	2.72	1.15	0.73	1.39	1.73	1.84	1.93	2.09
ZestXML	23.86	14.97	11.31	7.19	13.00	16.03	20.13	29.95
SentBERT	32.09	20.50	15.78	10.94	19.46	24.12	30.94	55.94
MPNet	34.58	22.02	16.86	11.96	21.32	26.30	33.53	57.78
MACLR (ours)	**44.27**	**28.46**	**21.83**	**15.14**	**27.04**	**33.33**	**42.03**	**67.95**
LF-Amazon-1M								
XR-Linear	0.51	0.20	0.12	0.36	0.42	0.43	0.45	0.49
Astec	0.49	0.59	0.12	0.34	0.40	0.42	0.44	0.49
SiameseXML	0.60	0.73	0.15	0.41	0.46	0.48	0.49	0.53
ZestXML	5.07	5.89	4.38	3.68	12.31	15.04	17.80	22.51
SentBERT	6.56	6.93	5.68	4.35	18.29	24.72	28.69	48.52
MPNet	8.87	10.34	7.56	6.78	20.11	26.14	31.98	50.48
MACLR (ours)	**10.37**	**11.23**	**8.58**	**7.57**	**23.55**	**29.60**	**36.71**	**56.44**

Table 3: Results of FS-XMC where the training subset covers 5% labels from the whole set.

Method	Precision			Recall				
	@1	@3	@5	@1	@3	@5	@10	@100
LF-Amazon-131K								
XR-Linear	5.09	2.09	1.32	2.36	2.86	3.02	3.18	3.74
Astec	3.94	1.92	1.26	2.31	3.34	3.66	4.00	4.96
SiameseXML	5.36	2.23	1.41	3.15	3.89	4.08	4.27	4.82
ZestXML	12.33	10.99	8.71	6.84	17.19	21.97	28.10	46.49
SentBERT	15.47	12.24	9.64	8.63	19.23	24.40	30.82	49.22
MPNet	15.03	11.88	9.28	8.47	18.74	23.69	29.93	48.84
MACLR (ours)	**19.56**	**16.19**	**12.64**	**11.15**	**25.65**	**32.18**	**39.63**	**58.45**
LF-WikiSeeAlso-320K								
XR-Linear	4.69	2.20	1.46	1.82	2.41	2.63	2.82	3.42
Astec	5.90	2.80	1.86	3.26	4.49	4.95	5.49	6.83
SiameseXML	6.83	3.15	2.06	3.88	5.15	5.56	6.02	7.09
ZestXML	10.06	8.11	6.60	5.33	11.49	14.74	19.57	40.46
SentBERT	18.47	14.19	11.29	10.82	21.55	26.77	33.92	57.02
MPNet	18.59	13.99	11.08	10.89	21.12	26.10	32.82	54.70
MACLR (ours)	**20.99**	**15.57**	**12.26**	**12.59**	**23.94**	**29.41**	**36.78**	**59.81**
LF-Wikipedia-500K								
XR-Linear	11.80	5.30	3.39	2.76	3.47	3.65	3.82	4.09
Astec	11.23	5.27	3.48	5.46	7.47	8.16	8.90	10.35
SiameseXML	12.44	5.69	3.79	6.05	7.98	8.62	9.22	10.40
ZestXML	27.31	17.31	13.09	8.28	15.13	18.64	23.30	36.50
SentBERT	41.06	26.35	20.25	14.17	25.34	31.32	39.77	66.24
MPNet	42.81	28.07	21.66	14.67	26.81	33.24	42.28	67.76
MACLR (ours)	**47.25**	**30.57**	**23.54**	**16.20**	**29.01**	**35.81**	**45.13**	**71.35**
LF-Amazon-1M								
XR-Linear	2.11	0.84	0.53	1.45	1.74	1.81	1.88	2.04
Astec	2.22	2.56	0.71	1.54	1.91	2.03	2.16	2.41
SiameseXML	2.60	3.01	1.06	1.81	2.20	2.30	2.41	2.60
ZestXML	7.17	8.35	6.36	5.18	17.49	21.88	26.80	36.51
SentBERT	8.89	10.02	7.93	7.00	21.58	27.35	33.98	54.28
MPNet	9.25	10.41	8.00	7.11	21.87	27.64	34.61	54.72
MACLR (ours)	**10.60**	**11.47**	**8.80**	**7.89**	**24.14**	**30.44**	**37.95**	**58.45**

tra supervision. However, MACLR is significantly better than MPNet with an average improvement of 3.5% in P@1 and over 10% in R@100. Furthermore, MACLR also outperforms its counterparts which are trained with effective pre-training tasks such as SimCSE and ICT on the target dataset, showing the effectiveness of pre-training strategies like multi-scale adaptive clustering in MACLR. Overall, results in Table 1 demonstrates that MACLR is capable to learn informative embeddings and to make useful predictions even with no supervision. We will investigate each component in MACLR in Section 5.4 thoroughly.

5.3 Few-Shot Learning

We further conduct few-shot learning (FS-XMC) experiments in which different learning algorithms can access a limited number of positive (instance, label) pairs. To simulate the scenario of few-shot learning, we first manually sample a small ratio of labels, then collect all their positive instances from the training set as the final subset of positive (instance, label) pairs for model training. Results of FS-XMC methods fine-tuned with 1% and 5% labels are shown in Tables 2 and 3 respectively.

Our proposed MACLR outperforms all other baselines significantly, including variants of Siamese-Transformer models (e.g., SentBERT, MPNet) and major competitive XMC methods (e.g., XR-Linear, Astec and SiameseXML), on all four datasets.

Note that SiameseXML is the state-of-the-art XMC method under the full supervision setup of XMC. Here, we again witness that existing XMC methods heavily rely on the supervision level as well as the full-coverage of label space for test set. MACLR, in contrast, still performs robustly under FS-XMC, which enjoy larger applicability to emerging domains with many cold-start labels.

Crucially, even ZestXML tailored to address the challenging scenario of unseen labels cannot match the performance of MACLR. In particular, when focusing on the few-shot scenario with only 1% sampled labels, MACLR achieves 18.74% in P@1, improving the performance of Astec with 0.94% and ZestXML with 10.10% significantly. Besides, MACLR outperforms all Sentence-BERT counterparts, validating the effectiveness of our pre-training procedure. As to fine-tuning on the subset with 5% labels, performance of all methods are improved as expected with more supervision.

The relative rank among these methods remains the same, with MACLR still performing the best in terms of precision and recall on all four datasets.

5.4 Ablation Study

In this part, we conduct an ablation study to investigate each component in our pre-training procedure, including multi-scale adaptive clustering, label regularization, and self-training with pseudo positive pairs constructed from the encoder or TF-IDF. We add a component once a time on LF-Amazon-131K to observe its independent influence on the model performance. Table 4 presents detailed performance on seven different configurations.

Table 4: Ablation study on LF-Amazon-131K.

Index	Ablation Configuration				Precision				Recall			
	MAC *	LR *	$\mathcal{E}$ †	TFIDF †	@1	@3	@5	@1	@3	@5	@10	@100
1	No	No	No	No	13.82	11.41	8.90	7.76	18.09	22.80	28.94	47.40
2	Yes	No	No	No	15.79	13.16	10.22	8.85	20.90	26.27	32.61	49.83
3	No	Yes	No	No	16.02	13.29	10.28	9.04	21.27	26.51	32.97	50.34
4	Yes	Yes	No	No	16.37	13.71	10.65	9.29	21.63	27.03	33.93	51.45
5	Yes	Yes	Yes	No	17.01	14.75	11.41	9.72	23.33	29.04	35.20	53.55
6	Yes	Yes	No	Yes	16.51	14.12	10.92	9.52	22.43	28.02	34.64	52.78
7	Yes	Yes	Yes	Yes	18.13	15.42	11.93	10.35	24.45	30.43	37.28	54.99

* MAC represents adaptive clustering while LR stands for label regularization.
† Pseudo positive pairs are constructed from $\mathcal{E}$ or TFIDF.

For two techniques multi-scale adaptive clustering and label regularization during the Stage I, they can improve the performance of the encoder separately, as shown in the performance gain of the index 2 and 3 over the index 1. When combined, they can further improve the accuracy of the model, from 8.90% to 10.65% in $P@5$ and from 47.40% to 51.45% in $R@100$. As to the second stage, we explore the impact of self-training with pseudo positive pairs either from the encoder itself or TF-IDF. We can see from Table 4 that pairs from both $\mathcal{E}$ and TF-IDF contribute to the precision and recall gain over the index 5. It further validates that the encoder and TF-IDF provides complementary perspective when constructing pseudo positive pairs.

6 Conclusions

This paper is the first to investigate the problem of Extreme zero-shot XMC without any supervision. We develop a two-stage pre-training procedure MACLR to train a Sentence-BERT style encoder on pseudo (context, title) pairs constructed from raw text. We demonstrate that techniques including multi-scale adaptive clustering, label regularization and self-training contribute to the performance gain of the pre-trained encoder. In particular, MACLR outperforms all unsupervised baselines significantly when there are no (instance, label) pairs provided. It also offers leading accuracy in both precision and recall after fine-tuning on a limited number of paired data. One limitation is relative low accuracy of top candidates and a future direction could be adding a ranker model after the encoder to improve performance on head labels.

Acknowledgement

Cho-Jui Hsieh is partially supported by NSF under IIS-2008173 and IIS-2048280.

Broader Impact

In the paper all datasets are publicly available without any private or confidential information and to the best of our knowledge, there are no ethical issues with this paper. For broader impacts of our work, we consider a practical scenario, Extreme Zero-Shot XMC where no supervision is required. Solving problems under EZ-XMC can reduce labor of manual annotation significantly. For example, the unsupervised model can help narrow the tagging space remarkably by selecting a small number of candidate labels for products in the e-commerce domain for efficient annotation. MACLR is such an effective method to benefit real-world applications like recommendation and advertisement.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM.

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. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 730–738.

Ilias Chalkidis, Emmanouil Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2019. Large-scale multi-label text classification on EU legislation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6314–6322, Florence, Italy. Association for Computational Linguistics.

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. 2021. 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, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020a. Pre-training tasks for embedding-based large-scale retrieval. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S. Dhillon. 2020b. Taming pretrained transformers for extreme multi-label text classification. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 3163–3171. ACM.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

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

Kunal Dahiya, Deepak Saini, Anshul Mittal, Ankush Shaw, Kushal Dave, Akshay Soni, Himanshu Jain, Sumeet Agarwal, and Manik Varma. 2021b. 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.

Ofer Dekel and Ohad Shamir. 2010. Multiclass-multilabel classification with more classes than examples. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 137–144. JMLR Workshop and Conference Proceedings.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Chuan Guo, Ali Mousavi, Xiang Wu, Daniel Niels Holtmann-Rice, Satyen Kale, Sashank J. Reddi, and Sanjiv Kumar. 2019. Breaking the glass ceiling for embedding-based classifiers for large output spaces. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4944–4954.

Ruiqi Guo, Sanjiv Kumar, Krzysztof Choromanski, and David Simcha. 2016. Quantization based fast inner product search. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, volume 51 of *JMLR Workshop and Conference Proceedings*, pages 482–490. JMLR.org.

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.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2020a. Revisiting self-training for neural sequence generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020b. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735. IEEE.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Contrastive pre-training for zero-shot information retrieval.

Himanshu Jain, Venkatesh Balasubramanian, Bhanu Chunduri, and Manik Varma. 2019. Slice: Scalable linear extreme classifiers trained on 100 million labels for related searches. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, pages 528–536. ACM.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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, Florence, Italy. Association for Computational Linguistics.

Yandong Li, Di Huang, Danfeng Qin, Liqiang Wang, and Boqing Gong. 2020. Improving object detection with selective self-supervised self-training. In *European Conference on Computer Vision*, pages 589–607. Springer.

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.

Tharun Medini, Qixuan Huang, Yiqiu Wang, Vijai Mohan, and Anshumali Shrivastava. 2019. Extreme classification in log memory using count-min sketch: A case study of amazon search with 50m products. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13244–13254.

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.

Aaron Mueller, Jason Krone, Salvatore Romeo, Saab Mansour, Elman Mansimov, Yi Zhang, and Dan Roth. 2022. Label semantic aware pre-training for few-shot text classification. *arXiv preprint arXiv:2204.07128*.

Thomas Müller, Guillermo Pérez-Torró, and Marc Franco-Salvador. 2022. Few-shot learning with siamese networks and label tuning. *arXiv preprint arXiv:2203.14655*.

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.

Yashoteja Prabhu, Anil Kag, Shrutendra Harsola, Rahul Agrawal, and Manik Varma. 2018. Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 993–1002. ACM.

Yashoteja Prabhu and Manik Varma. 2014. Fastxml: a fast, accurate and stable tree-classifier for extreme multi-label learning. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 263–272. ACM.

Anand Rajaraman and Jeffrey David Ullman. 2011. *Mining of massive datasets*. Cambridge University Press.

Sashank J. Reddi, Satyen Kale, Felix X. Yu, Daniel Niels Holtmann-Rice, Jiecao Chen, and Sanjiv Kumar. 2019. Stochastic negative mining for learning with large output spaces. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, 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, Hong Kong, China. Association for Computational Linguistics.

Anshumali Shrivastava and Ping Li. 2014. Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2321–2329.

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

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Qizhe Xie, Minh-Thang Luong, Eduard H. Hovy, and Quoc V. Le. 2020. Self-training with noisy student improves imagenet classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10684–10695. IEEE.

Nishant Yadav, Rajat Sen, Daniel N. Hill, Arya Mazumdar, and Inderjit S. Dhillon. 2021. Session-aware query auto-completion using extreme multi-label ranking. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 3835–3844, New York, NY, USA. Association for Computing Machinery.

I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. 2019. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*.

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 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5812–5822.

Hsiang-Fu Yu, Kai Zhong, Jiong Zhang, Wei-Cheng Chang, and Inderjit S Dhillon. 2020. PECOS: Prediction for enormous and correlated output spaces. *Journal of Machine Learning Research*.

Jiong Zhang, Wei-cheng Chang, Hsiang-fu Yu, and Inderjit S Dhillon. 2021. Fast multi-resolution transformer fine-tuning for extreme multi-label text classification. In *Advances in Neural Information Processing Systems*.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. 2020. Rethinking pre-training and self-training. *Advances in neural information processing systems*, 33:3833–3845.

A Implementation Details

Datasets and Source Codes Statistics of four datasets are presented in Table 5. All XMC datasets used in the paper are available in the Extreme Classification Repository (Bhatia et al., 2016)[1], except for LF-Amazon-1M, which is available from the ZestXML paper (Gupta et al., 2021)[2]. Besides, sources code can be accessed in this link.

Table 5: Dataset statistics. $\mathbf{N}_{\text{train}}$, $\mathbf{N}_{\text{test}}$ and $\mathbf{N}_{\text{label}}$ are the number of training points, test points, and labels respectively. $\mathbf{D}_{\text{BoW}}$ is the dimensionality of Bag-of-Words (BoW) features.

Dataset	$\mathbf{N}_{\text{train}}$	$\mathbf{N}_{\text{test}}$	$\mathbf{N}_{\text{label}}$	$\mathbf{D}_{\text{BoW}}$
LF-Amazon-131K	294,805	134,835	131,073	80,000
LF-WikiSeeAlso-320K	693,082	177,515	312,330	80,000
LF-Wikipedia-500K	1,813,391	783,743	501,070	500,000
LF-Amazon-1M	914,179	1,465,767	960,106	1,000,000

Compared Baselines of EZ-XMC Here we provide detailed description of each baseline method under the EZ-XMC setting.

- TF-IDF (Rajaraman and Ullman, 2011), which represents instances and labels by sparse TF-IDF features and retrieves top labels for each instance based on the similarity of TF-IDF features;
- XR-Linear (Yu et al., 2020), a hierarchical linear model trained with pseudo positive pairs constructed from TF-IDF;
- GloVe (Pennington et al., 2014), which adopts dense average word embeddings with the dimension of 300 trained on co-occurrence statistics to measure similarity between instances and labels;
- Sentence-BERT (SentBERT) (Devlin et al., 2019; Reimers and Gurevych, 2019), a sentence encoder modeled as a Siamese-Transformer to derive semantically meaningful embeddings for instances and labels;
- Paraphrase MPNet (MPNet) (Song et al., 2020), another Sentence-BERT model originally designed for searching sentence paraphrases;
- SimCSE (Gao et al., 2021), a Siamese-Transformer pre-trained with the contrastive objective using dropout noise as augmentation;
- ICT (Lee et al., 2019), another Siamese-Transformer pre-trained with the contrastive objective using (context, title) pairs.

Evaluation Metrics As mentioned before, we adopt precision and recall as our evaluation metrics. In detail, $P@k$ and $R@k$ are defined as follows:

$$P@k = \frac{1}{k} \sum_{i \in \text{rank}_k(y')} y_i, \quad R@k = \frac{1}{\sum_l y_l} \sum_{i \in \text{rank}_k(y')} y_i. \tag{5}$$

$y \in \{0,1\}^L$ and $y' \in R^L$ are the ground truth vector and the prediction vector respectively. rank_k returns the indices of the top-k highest elements.

Hyper-parameters We use a Siamese Transformer model to embed both instances and labels. The encoder consists of a 12 layers BERT-base model, topped with a linear head projecting hidden state of the [CLS] token into a 512-dimensional embedding. The sequence length of the instance and the label is set to be 288 and 64 respectively. We pre-train the model on eight V100 GPUs for 100,000 steps with an Adam optimizer and batch size of 32 per GPU in both Stage I and Stage II. This pre-training process takes about 1 day. We adopt an initial learning rate 1×10^{-5} with the warm-up ratio 0.1, followed by a linear learning rate decay. For fine-tuning, the learning rate of Adam is set to 5×10^{-6} with 2000 training steps for the 1% label ratio and 10K training steps for

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

[2] https://github.com/nilesh2797/zestxml

Table 6: Mean and variance of MACLR performance of four independent runs under EZ-XMC on LF-Amazon-131K.

Method	Precision			Recall				
	@1	@5	@10	@1	@3	@5	@10	@100
MACLR	17.64 ± 0.11	15.24 ± 0.01	11.81 ± 0.01	10.11 ± 0.04	24.13 ± 0.04	30.14 ± 0.04	37.13 ± 0.03	54.88 ± 0.01

the 5% label ratio. In the Stage I, we use the initial cluster size $K_0 = 2048$ and set $T_K = 10000$ and $T_{\text{update}} = 5000$. In Stage II, top 3 ranked labels from predictions of the encoder and TF-IDF are selected to constitute the pseudo set for self-training.

For hyper-parameters of all baselines, we follow their default setups. All experiments are conducted on the AWS p3dn.24xlarge instance, consisting of 96 Intenl Xeon CPUs with 768 GB of RAM and 8 Nvidia V100 GPUs with 32 GB of memory each. It takes about half a day to complete the pre-training procedure of MACLR.

We present error bars of four independent runs to validate our MACLR results are statistically significant under EZ-XMC in Table 6. It can be observed that the variance is small, showing that our method can produce similar results with different random seeds, and MACLR is statically better than other baselines compared with results in Table 1. Therefore, we run each method for four times and report the best performance in the main paper.

B Hyperparameter Study

We provide details of hyperparameter tuning process. For hyperparameters in the Stage I including T_K, T_{update}, and K_0, we choose them from the following ranges: $T_K \in [5000, 10000, 25000]$, $T_{\text{update}} \in [1000, 5000, 10000]$, and $K_0 \in [1024, 2048, 4096]$. Instead of grid search for all possible combinations (in total $3^3 = 27$), we tried to investigate the impact of different hyperparameters first and then determined the value. Specifically, we choose the median setting of ($T_K = 10000, T_{\text{update}} = 5000, K_0 = 2048$) as the baseline, and for each hyperparameter to be evaluated, we fix the remaining two and observe the performance. Experimental results can be found in Table 7, 8 and 9. We summarize our findings below:

- With $T_{\text{update}} = 5000, K_0 = 2048, T_K = 10000$ performs the best. We assume that when T_K is smaller as 5000, the training steps are not sufficient to obtain satisfactory representations under the current cluster size; and when T_K is larger as 25000, the algorithm spends too much time on one cluster size and fails to explore different scenarios based on curriculum learning.
- With $T_K = 10000, K_0 = 2048, T_{\text{update}} = 1000$ outperforms the other two values. It is expected since the cluster assignment would be more accurate and the learned embeddings would be better accordingly.
- With $T_K = 10000, T_{\text{update}} = 5000, K_0 = 2048$ is the best among three values. We hypothesize the trajectory of $K_0 = 2048$ gives the appropriate curriculum learning process.

Based on these observations, the best setting should be ($T_K = 10000, T_{\text{update}} = 1000, K_0 = 2048$). However, we can see that $T_{\text{update}} = 1000$ is only slightly better than $T_{\text{update}} = 5000$ and considering the cost of updating cluster assignment, we select $T_{\text{update}} = 5000$ instead.

Table 7: Impact of T_K with fixed $T_{\text{update}} = 5000$ and $K_0 = 2048$ on LF-Amazon-131K.

Hyperparameter	Precision			Recall				
	@1	@3	@5	@1	@3	@5	@10	@100
$T_K = 5000$	15.35	12.78	9.67	8.22	20.56	26.20	33.03	50.24
$T_K = 10000$	**16.37**	**13.71**	**10.65**	**9.29**	**21.63**	**27.03**	**33.93**	**51.45**
$T_K = 25000$	14.46	12.05	9.24	7.68	19.96	25.78	32.64	49.16

Table 8: Impact of T_{update} with fixed $T_K = 10000$ and $K_0 = 2048$ on LF-Amazon-131K.

Hyperparameter	Precision			Recall				
	@1	@3	@5	@1	@3	@5	@10	@100
$T_{\text{update}} = 1000$	**16.61**	**13.82**	**10.90**	**9.63**	**21.72**	**27.35**	**34.32**	**51.82**
$T_{\text{update}} = 5000$	16.37	13.71	10.65	9.29	21.63	27.03	33.93	51.45
$T_{\text{update}} = 10000$	15.26	12.66	9.72	8.42	20.59	26.14	32.98	50.04

Table 9: Impact of K_0 with fixed $T_{\text{update}} = 5000$ and $T_K = 10000$ on LF-Amazon-131K.

Hyperparameter	Precision			Recall				
	@1	@3	@5	@1	@3	@5	@10	@100
$K_0 = 1024$	15.82	13.03	9.94	8.74	20.85	26.40	33.06	50.28
$K_0 = 2048$	**16.37**	**13.71**	**10.65**	**9.29**	**21.63**	**27.03**	**33.93**	**51.45**
$K_0 = 4096$	15.98	13.24	10.05	8.87	21.14	26.69	33.36	50.62

For the hyperparameter k in the Stage II, we use the previous setting ($T_K = 10000, T_{\text{update}} = 5000, K_0 = 2048$) for pretraining in the stage I, and tune k from $[1, 3, 5]$. Results are shown in

Table 10 in the paper. It can be observed that $k = 3$ leads to the best performance. This is due to the fact that $k = 1$ did not introduce a sufficient number of meaningful pseudo (instance, label) pairs while $k = 5$ was likely to add to the noise of the training set with irrelevant (instance, label) pairs.

Table 10: Impact of the number of pseudo labels k on LF-Amazon-131K.

Hyperparameter	Precision			Recall				
	@1	@3	@5	@1	@3	@5	@10	@100
$k = 1$	17.15	14.92	11.38	9.72	23.76	29.55	36.68	54.46
$k = 3$	**18.13**	**15.42**	**11.93**	**10.35**	**24.45**	**30.43**	**37.28**	**54.99**
$k = 5$	16.74	14.06	10.93	9.48	23.51	29.12	36.19	53.98

Therefore, the default hyperparameter choice of MACLR is determined as ($T_K = 10000, T_{\text{update}} = 5000, K_0 = 2048, k = 3$), which is tuned on LF-Amazon-131K. Moreover, as we can see that MACLR is not very sensitive to the choice of hyperparameters and performs better than all the baselines, we assume that this configuration would perform similarly on other datasets and use it without further tuning.

C Additional Experiments on FS-XMC

Table 11: Results of FS-XMC where the training subset covers 1% positive pairs from the whole set.

Method	Precision			Recall				
	@1	@3	@5	@1	@3	@5	@10	@100
LF-Amazon-131K								
XR-Linear	5.37	2.66	1.68	2.81	3.92	4.09	4.26	4.99
Astec	3.29	2.04	1.41	1.93	3.33	3.77	4.06	5.06
SiameseXML	7.14	3.74	2.41	4.22	6.17	6.55	6.95	8.09
ZestXML	12.91	11.31	8.91	7.20	17.69	22.51	28.27	42.40
SentBERT	15.08	11.81	9.06	8.38	18.42	22.89	28.62	46.38
MPNet	15.26	12.30	9.42	8.56	19.35	23.98	29.91	48.06
MACLR (ours)	**18.92**	**16.17**	**12.62**	**10.98**	**25.64**	**32.16**	**39.46**	**58.24**
LF-WikiSeeAlso-320K								
XR-Linear	6.97	3.43	2.31	3.74	5.02	5.44	5.84	6.87
Astec	5.58	3.35	2.48	3.22	5.43	6.51	7.95	11.76
SiameseXML	9.87	5.22	3.59	5.84	8.57	9.53	10.60	13.04
ZestXML	10.40	8.18	6.49	5.57	11.65	14.52	18.81	33.20
SentBERT	18.85	14.23	11.22	11.16	21.77	26.94	33.78	55.88
MPNet	18.04	13.27	10.44	10.51	19.99	24.62	30.86	52.52
MACLR (ours)	**20.49**	**15.50**	**12.24**	**12.34**	**23.88**	**29.43**	**36.76**	**59.82**

In this section, we present additional experimental results for the setting of FS-XMC on LF-Amazon-131K and LF-WikiSeeAlso-320K. Instead of sampling a few-shot subset by the label coverage ratio, we turn to sampling based on the pair ratio. Specifically, suppose a training set $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}$ has $|\mathcal{D}_{\text{train}}|$ positive pairs. Each time we randomly sample a small ratio of δ (1% or 5% in our paper) pairs from the total set to constitute the few-shot subset. Each subset has $\delta|\mathcal{D}_{\text{train}}|$ pairs for fine-tuning. Detailed results of $\delta = 1\%$ and $\delta = 5\%$ are presented in Table 11 and 12 respectively. MACLR is still the best-performing method and outperforms all other baselines significantly in precision and recall.

Table 12: Results of FS-XMC where the training subset covers 5% positive pairs from the whole set.

Method	Precision			Recall				
	@1	@3	@5	@1	@3	@5	@10	@100
LF-Amazon-131K								
XR-Linear	11.20	5.82	3.80	5.98	8.56	9.18	9.80	12.79
Astec	10.71	6.50	4.52	6.12	10.23	11.67	13.35	18.15
SiameseXML	11.88	8.72	5.93	8.50	13.68	15.23	16.80	20.28
ZestXML	12.86	11.28	8.91	7.10	17.62	22.43	28.42	49.41
SentBERT	16.94	13.59	10.52	9.55	21.23	26.55	33.14	51.81
MPNet	17.48	13.58	10.61	9.95	21.38	26.83	33.60	52.31
MACLR (ours)	**19.75**	**16.45**	**12.87**	**11.18**	**25.99**	**32.70**	**40.38**	**59.82**
LF-WikiSeeAlso-320K								
XR-Linear	13.13	6.88	4.70	7.00	9.64	10.54	11.49	14.20
Astec	15.61	8.73	6.23	8.77	13.17	15.02	17.36	24.30
SiameseXML	16.51	9.68	6.96	9.40	14.78	16.97	19.48	25.26
ZestXML	17.68	8.51	6.85	10.63	12.01	15.20	20.08	43.10
SentBERT	20.12	15.01	11.87	12.05	23.01	28.40	35.52	58.41
MPNet	19.88	14.90	11.76	11.85	22.75	27.96	35.03	57.26
MACLR (ours)	**21.80**	**16.61**	**13.12**	**13.27**	**25.74**	**31.59**	**39.25**	**62.13**