Self-Guided Unsupervised Contrastive Learning for Extreme Zero-Shot Text Classification

Anonymous ACL submission

Abstract

Extreme zero-shot multi-label text classification (ZXMTC) is the task of retrieving relevant labels from extremely large label spaces without any annotated document–label pairs. Prior work mainly relies on lexical matching or self-supervised contrastive learning, often depend on shallow lexical signals and noisy pseudo labels. We propose Curriculum Learning with Adversarial Self-training and Progressive Optimization (CLASP), a contrastive framework including Label-Aware Cloze Task (LACT) as an adversarial method to create pseudo-label-oriented pseudo pairs, encouraging learning richer semantic representations. CLASP progressively shifts from adversarial cloze-training to pseudo-label self-training. Experiments on large-scale ZXMTC benchmarks show that CLASP consistently improves over strong baselines and achieves the best average early-rank retrieval performance with a compact pre-trained encoder.

1 Introduction

Extreme multi-label text classification (XMTC) assigns relevant labels to a text from an extremely large label space (thousands or millions), introducing major challenges: highly imbalanced label distributions and efficient training and inference.

Early approaches: Parabel (Prabhu et al., 2018), Bonsai (Khandagale et al., 2019), XR-Linear (Yu et al., 2022), AttentionXML (You et al., 2019) construct hierarchical partitions to reduce the effective search space. Later, transformer-based X-Transformer (Chang et al., 2020), LightXML (Jiang et al., 2021), XR-Transformer (Zhang et al., 2021) successively fine-tune on multi-resolution objectives. These methods assume full supervision with annotated (instance, label) pairs, an assumption often unrealistic. Moreover, supervised methods ignore label semantics, despite the informative nature of label descriptions.

Recent research shift toward zero-shot extreme multi-label text classification (ZXMTC) with no (instance, label) pairs available during training. The model has access only to raw documents and label descriptions. ZXMTC is commonly formulated as semantic retrieval, where an encoder $\mathcal{E}$ maps both documents and labels into a shared embedding space for nearest-neighbor search. ZXMTC remains challenging, demands aligning label descriptions with long documents without supervision while handling millions of embeddings efficiently. Existing approaches can be grouped as follows.

Lexical matching such as TF-IDF or BM25 (Robertson and Zaragoza, 2009) rely on lexical overlap between documents and labels. They remain strong baselines with minimal computation.

Semantic baselines. Pre-trained encoders (Sent-BERT (Reimers and Gurevych, 2019), MPNet (Song et al., 2020)) provide semantically meaningful embeddings. However, general-purpose encoders need refinement to improve alignment.

Contrastive representation learning. Models learn to bring closer positive pairs while contrasting them with sampled negatives, making large-scale training feasible. In ZXMTC, pseudo pairs come from self-training (Zoph et al., 2020; Xie et al., 2020; He et al., 2020); text partitioning DeCLUTR (Giorgi et al., 2021), RTS (Zhang et al., 2022), or from document structure ICT (Lee et al., 2019) and MACLR (Xiong et al., 2022). These approaches enforce similar embeddings between augmented views of the same input, however rely on predefined document structures or random sampling.

In this work, we propose CLASP, an adversarial pseudo-label-guided contrastive framework that constructs training pairs by removing informative segments associated with pseudo-label predictions. Experiments show that our method consistently improves both the base encoder and standard self-training approaches. Combined with a pre-trained encoder backbone, our framework achieves state-

of-the-art performance on ZXMTC benchmarks on average, even against LLMs-based approaches.

2 Proposed method

2.1 Problem formulation

Let $X = \{x_i\}$ be a set of documents and $Y = \{y_j\}$ be the set of labels, where each $y_j \in Y$ can be a textual description of each label. In representation learning, we aim to train an encoder $\mathcal{E}$ that maps texts, documents x_i or labels y_i, into the same latent space, so that semantically similar inputs yeild similar embeddings. Predictions are made with nearest-neighborhood search in this latent space:

$$\mathrm{s}_{\mathcal{E}}(x_i, y_j) = \exp\left(\mathcal{E}(x_i) \cdot \mathcal{E}(x_j)/\tau\right) \quad (1)$$

2.2 CLASP algorithm

2.2.1 Pre-training with ICT

Inversed cloze task (ICT) (Lee et al., 2019) proposes a cloze-based approach: randomly select a sentence as pseudo positive label and the rest of the document as the pseudo instance. Particularly, as the document's title captures its meaning, one can use the (content, title) pairs. For a document with m sentences $x_i = (s_{i1}, s_{i2}, ..., s_{im})$, let $\bar{x}_i = (s_{i2}, ..., s_{im})$, and $\bar{y}_i = s_{i1}$ being its title. The ICT objective is:

$$\mathcal{L}_{\mathrm{ICT}} = \sum_{i=1}^{N} -\log \frac{\mathrm{s}_{\mathcal{E}}\left(\bar{x}_i, \bar{y}_i\right))}{\sum_{j=1}^{N} \mathrm{s}_{\mathcal{E}}\left(\bar{x}_i, \bar{y}_j\right))} \quad (2)$$

2.2.2 Label-aware cloze task (LACT)

Generally, self-training (ST) implies training the model with pseudo positive pairs generated by itself. For pseudo positive pairs $\{(x_i, \hat{y}_i)\}$:

$$\mathcal{L}_{\mathrm{ST}} = \sum_{i=1}^{N} \frac{\sum_{j=1}^{N} -\delta_{ij} \log \frac{\mathrm{s}_{\mathcal{E}}(x_i, \hat{y}_j)}{\sum_{j'=1}^{N} \mathrm{s}_{\mathcal{E}}\left(x_i, \hat{y}_{j'}\right)}}{\sum_{j=1}^{N} \delta_{ij}} \quad (3)$$

Where $\delta_{ij} = \mathbb{1}_{\hat{y}_i = \hat{y}_j}$ indicates instances with same pseudo labels.

Instead of using a random partitioning (RTS (Zhang et al., 2022)) or document title (ICT (Lee et al., 2019)) as pseudo labels, we propose an adversarial approach: **Label-aware cloze task (LACT)**. Observing that naive self-training adds no new information and instead may amplify noise, we deliberately corrupt the $(x_i, \hat{y}_i)$ pair to create a more challenging training. Given a document $x_i = (s_{i1}, s_{i2}, ..., s_{im})$, for which $\hat{y}_i$ is a predicted pseudo label. The most similar sentence to $\hat{y}_i$, called *key sentence*, is identified based on a scorer g:

$$k_i = \arg\max_j g(s_{ij}, \hat{y}_i) \quad (4)$$

Naturally, g could be based on $\mathcal{E}$, e.g. $g = s_{\mathcal{E}}$. However, extensively embedding all sentences of every x during training is costly, especially with long documents. Introducing an independent model here may induce additional useful information. Therefor, among strong light-weight encoders, we utilize TF-IDF, trained on the learning corpus.

To prevent trivial learning (as s_{ik_i} is literally in x_i), we remove s_{ik_i} from x to create the corrupted input $\tilde{x}_i = x_i \backslash \{s_{ik_i}\}$. The LACT loss is adapted from equation (3), replacing $(x_i, \hat{y}_i)$ with $(\tilde{x}_i, s_{ik_i})$:

$$\mathcal{L}_{\mathrm{LACT}} = \sum_{i=1}^{N} \frac{\sum_{j=1}^{N} -\delta_{ij} \log \frac{\mathrm{s}_{\mathcal{E}}\left(\tilde{x}_i, s_{jk_j}\right)}{\sum_{j'=1}^{N} \mathrm{s}_{\mathcal{E}}\left(\tilde{x}_i, s_{j'k_j}\right)}}{\sum_{j=1}^{N} \delta_{ij}} \quad (5)$$

2.2.3 Progressive multi-task: LACT and ST

Training starts with a label-oriented cloze task with $\mathcal{L}_{\mathrm{LACT}}(\tilde{x}_i, s_{ik_i})$ and then gradually shifts to self-training with $\mathcal{L}_{\mathrm{ST}}(\tilde{x}_i, \hat{y}_i)$. Our training loss combines both LACT and ST, with a linear scheduler α from 1 to 0:

$$\mathcal{L} = (1-\alpha)\mathcal{L}_{\mathrm{ST}}(\tilde{x}_i, \hat{y}_i) + \alpha\mathcal{L}_{\mathrm{LACT}}(\tilde{x}_i, s_{ik_i}) \quad (6)$$

Training with $\mathcal{L}_{\mathrm{LACT}}(\tilde{x}_i, s_{ik_i})$ fallbacks to matching two segments of a document. As $\hat{y}_i$ is not evolved, the error when predicting $\hat{y}_i$ will not propagate in future training. Training with $\mathcal{L}_{\mathrm{ST}}(\tilde{x}_i, \hat{y}_i)$

Algorithm 1: Curriculum Learning with Adversarial Self-training and Progressive Optimization (CLASP)

Input: Encoder $\mathcal{E}$, documents X, parameters T, T_1, T_2, scoring function g
Output: Trained encoder $\mathcal{E}$
Use $\mathcal{E}$ to retrieve $\hat{y}$ for all $x \in X$;
for $t \leftarrow 1$ **to** T **do**
  Sample a batch $\{x_i\} \subset X$;
  if $t \leq T_1$ **then** // ICT
    $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{ICT}}(\bar{x}_i, \bar{y}_i)$;
  else if $t \leq T_1 + T_2$ **then** // LACT+selftrain
    $\alpha_t \rightarrow \frac{T_1+T_2-t}{T_2}$;
    Stochastic approximation $\mathcal{L} \leftarrow (1-\alpha_t)\mathcal{L}_{\mathrm{ST}}(\tilde{x}_i, \hat{y}_i) + \alpha_t\mathcal{L}_{\mathrm{LACT}}(\tilde{x}_i, s_{ik_i})$;
  else // selftrain
    $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{ST}}(x_i, \hat{y}_i)$;
  end
  $\mathcal{L} \leftarrow \mathcal{L} + \mathcal{L}_{\mathrm{reg}}$;
  Train $\mathcal{E}$ by minimizing $\mathcal{L}$;
end

Method	P@1	P@3	P@5	R@1	R@3	R@5	R@10	P@1	P@3	P@5	R@1	R@3	R@5	R@10
	LF-Amazon-131K							**LF-WikiSeeAlso-320K**						
TF-IDF	12.38	11.50	9.14	6.91	18.14	23.21	29.32	10.71	8.90	7.15	5.92	13.03	16.48	21.60
BM25	17.4	11.8	8.8	10.0	18.5	22.4	26.8	18.5	12.0	9.0	10.1	17.5	21.0	
GloVe	3.67	2.78	2.15	2.05	4.33	5.44	7.23	3.86	2.76	2.21	2.12	4.11	5.22	6.95
SentBERT	1.86	1.44	1.14	1.01	2.22	2.88	4.01	1.71	1.27	1.06	1.08	2.16	2.90	4.17
SimCSE	10.13	8.61	6.69	5.61	13.39	16.84	21.27	9.03	6.64	5.22	4.99	9.89	12.34	15.93
MPNet	13.94	11.41	8.82	7.82	18.08	22.58	27.91	13.75	11.93	9.58	8.14	17.77	22.21	28.11
ICT	13.82	11.41	8.90	7.76	18.09	22.80	28.94	10.76	10.05	8.12	6.12	14.32	18.05	23.01
TAS-B	13.5	12.3	9.6	8.1	20.3	25.5	31.3	23.7	**16.1**	**12.5**	13.1	**23.8**	**29.2**	**36.5**
MACLR	18.13	**15.42**	11.93	10.35	24.45	30.43	37.28	16.31	13.53	10.78	9.71	20.39	25.37	32.05
RTS	18.74	15.30	**11.96**	10.64	24.16	30.45	**38.19**	18.64	15.14	12.07	10.86	22.68	28.29	35.47
GPT3.5	17.1	11.0	8.4	9.7	17.7	21.9	27.4	**24.6**	15.6	11.6	**13.3**	22.8	27.1	32.7
ICXML	**22.5**	14.8	10.9	**14.1**	**26.6**	**32.0**	34.9	24.1	15.0	10.9	12.8	20.1	24.2	30.1
CLASP	21.00	14.26	10.66	11.94	22.47	27.18	33.42	23.89	15.68	11.98	13.15	22.93	27.73	34.19
	LF-Wikipedia-500K							**LF-Amazon-1M**						
TF-IDF	20.30	12.98	9.96	7.25	12.91	15.98	20.31	7.68	9.20	7.23	5.61	19.30	24.92	31.76
GloVe	2.19	1.52	1.23	0.85	1.66	2.18	3.1	4.05	4.07	3.07	2.91	8.42	10.44	12.90
SentBERT	0.17	0.15	0.13	0.05	0.13	0.18	0.30	2.82	2.87	2.13	2.03	5.91	7.21	8.80
SimCSE	14.32	6.84	4.55	4.24	8.03	11.26	14.35	3.33	3.69	2.74	2.38	7.66	9.38	11.43
MPNet	22.46	12.87	9.49	8.74	14.07	16.76	20.64	8.29	8.87	6.80	6.04	18.64	23.51	29.35
ICT	17.74	9.67	7.06	7.35	11.60	13.84	17.19	8.66	9.26	7.13	6.30	19.45	24.60	30.73
MACLR	28.44	17.75	13.53	10.40	18.16	22.38	28.52	9.58	10.41	8.03	7.38	22.01	27.72	34.48
RTS	30.67	19.03	14.34	10.58	18.48	22.51	28.23	10.00	**10.95**	**8.41**	7.34	**23.09**	**29.14**	**36.30**
CLASP	**37.13**	**21.37**	**15.65**	**12.95**	**21.02**	**24.96**	**30.38**	**17.36**	10.78	7.88	**12.64**	22.53	27.18	33.24

Table 1: ZXMTC evaluation on four datasets. **Bold** is the best, underline are the second and third.

provides additional signals than $\mathcal{L}_{\mathrm{ST}}(x_i, \hat{y}_i)$. As g (TF-IDF) is lexical matching, this encourages exploiting additional semantic signals than relying on trivial lexical cues.

In equation (6), encoding $\tilde{x}_i, \hat{y}_i, \tilde{x}_i$ and s_{ik_i} increases computational cost, we adopt a stochastic approximation strategy instead. For each x, $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{ST}}(\tilde{x}_i, \hat{y}_i)$ with probability $1 - \alpha_t$ and $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{LACT}}(\tilde{x}_i, s_{ik_i})$ with probability α_t. Beyond reducing computation, this naturally integrates with the proposed curriculum strategy, where α_t progressively control the transition.

2.2.4 Regularization

(Gao et al., 2021) confirms that dropout acts as a data augmentation. If the same instance x_i goes through $\mathcal{E}$ twice, the two slightly different embeddings $\mathcal{E}(x_i)$ and $\mathcal{E}(x_i^+)$ should have similar semantics. Also, randomly sampled labels $\{y_j^{\mathrm{reg}}\}$ from Y are considered negative labels. Thus, a regularization term is added:

$$\mathcal{L}_{\mathrm{reg}} = \sum_{i=1}^{N} -\log \frac{\mathrm{s}_{\mathcal{E}}\left(x_i, x_i^+\right)}{\mathrm{s}_{\mathcal{E}}\left(x_i, x_i^+\right) + \sum_j \mathrm{s}_{\mathcal{E}}\left(x_i, y_j^{\mathrm{reg}}\right)} \tag{7}$$

Minimizing $\mathcal{L}_{\mathrm{reg}}$ enforces that embedding displacement by dropout is minimal compared to the distance to random labels.

2.2.5 Summary of the CLASP algorithm

In the current implementation, All-MiniLM-L12-v2[1] pre-trained contrastively on sentence semantics, is chosen as our backbone encoder due to its compact size (see appendix F).

The training procedure is summarized in Algorithm 1, consisting of three stages: ICT, progressive mixing LACT and ST, and ST (implementation details in appendix E).

3 Experiments

3.1 Data, baselines and evaluation

We evaluate on four public ZXMTC datasets (Bhatia et al., 2016; Gupta et al., 2021) derived from Amazon recommendation and Wikipedia articles, with label sets ranging from 131K to 1M in size. Details are presented in table 4.

Results of baselines for ZXMTC are extracted from (Zhang et al., 2022; Zhu and Zamani, 2024). More details on these approches can be found in appendix B. Where possible, the methods trained on unsupervised data and the label set are then evaluated on the same test set. We evaluate with information retrieval metrics over the top-k labels retrieved by embedding similarity: Precision@k (P@k) and Recall@k (R@k) (see appendix A).

[1](Wang et al., 2020).https://www.sbert.net/docs/sentence_transformer/pretrained_models.html

Method	P@1	P@3	P@5	R@1	R@3	R@5	R@10
TF-IDF	12.77	10.65	8.37	6.42	15.85	20.15	25.75
GloVe	3.44	2.78	2.17	1.98	4.63	5.82	7.55
SentBERT	1.64	1.43	1.12	1.04	2.61	3.29	4.32
SimCSE	9.20	6.45	4.80	4.31	9.74	12.46	15.75
MPNet	14.61	11.27	8.67	7.69	17.14	21.27	26.50
ICT	12.75	10.10	7.80	6.88	15.87	19.82	24.97
MACLR	18.12	14.28	11.07	9.46	21.25	26.48	33.08
RTS	19.51	15.11	**11.70**	9.86	22.10	**27.60**	**34.55**
CLASP	**24.85**	**15.52**	11.54	**12.67**	**22.24**	26.76	32.81

Table 2: ZXMTC evaluation averaged across datasets. **Bold** is the best, underline are the second and third.

3.2 Main results

Tables 1 and 2 summarizes the metrics from unsupervised techniques on four ZXMTC, including our proposed method.

Among lexical matching approaches, TF-IDF and BM25 remain strong baselines, outperforming many BERT variants (SentBERT, SimCSE, ICT). Among SentBERT variants, only MPNet is comparable to TF-IDF. MACLR and RTS are consistently strong and similar across datasets. They outperform even LLM-based methods like vanilla GPT3.5 and ICXML on two datasets.

CLASP achieves the best average performance across P@1, P@3, R@1, and R@3, indicating strong early-rank accuracy, especially on LF-Wikipedia-500K. On LF-Amazon-1M, CLASP substantially improves over dense baselines, suggesting good scalability to very large label spaces. However, for later ranks as R@10 and R@100, our method slightly lags behind RTS and MACLR. Nevertheless, from table 5, ours uses a smaller model than these two baselines.

Notably, results on LF-Amazon-131K and LF-WikiSeeAlso-320K show that representation contrastive learning methods on small sentence transformers, such as MACLR, RTS, and CLASP, exhibit competitive compared to free GPT-3.5 generation or ICXML, despite the massive size difference ($\sim$ 1,500 times, table 5)[2]. Strong specialized sparse/dense retrievers still outperform general LLMs, particularly for large label spaces. LLMs excel at few-shot top predictions but struggle with long-tail coverage.

3.3 Ablation Studies

Table 3 shows results obtained with different configurations of our algorithm CLASP, including the variant CLASP+ where the pseudo labels $\hat{y}_i$ is retrieved after each stage[3]. The scores are algebraic gain compared to the performance of the base model All-MiniLM-L12-v2 (appendix F).

Base model	P@1	P@3	P@5	R@1	R@3	R@5	R@10
LF-Amazon-131K							
Self-training	-0.73	-0.10	-0.04	-0.51	-0.24	-0.98	+0.26
CLASP+	**+1.39**	**+1.28**	**+0.95**	**+0.76**	**+1.9**	**+1.49**	**+2.82**
CLASP	+0.55	+0.59	+0.45	+0.29	+0.89	+0.29	+1.76
CLASP w/o ICT	-0.01	-0.05	-0.05	-0.06	-0.10	-0.99	+0.13
CLASP w/o LACT	+0.34	+0.38	+0.35	+0.14	+0.57	+0.06	+1.28
LF-WikiSeeAlso-320K							
Self-training	-0.04	+0.07	+0.06	-0.05	+0.06	+0.10	+0.19
CLASP+	-6.31	-3.94	-3.02	-3.79	-6.58	-7.98	-10.02
CLASP	+0.46	+0.34	+0.22	+0.34	+0.52	+0.58	+0.61
CLASP w/o ICT	**+0.50**	**+0.42**	**+0.29**	**+0.35**	**+0.65**	**+0.72**	**+0.81**
CLASP w/o LACT	+0.45	+0.31	+0.21	+0.30	+0.47	+0.53	+0.60
LF-Wikipedia-500K							
Self-training	+5.20	+2.52	+1.70	+1.58	+2.09	+2.27	+2.40
CLASP+	-12.29	-8.28	-6.41	-3.31	-6.43	-8.07	-10.46
CLASP	+6.86	+3.32	+2.25	+2.17	+2.86	+3.12	+3.33
CLASP w/o ICT	**+7.08**	**+3.50**	**+2.39**	**+2.19**	**+3.01**	**+3.31**	**+3.61**
CLASP w/o LACT	+6.83	+3.33	+2.27	+2.12	+2.82	+3.12	+3.37
LF-Amazon-1M							
Self-training	-0.29	-0.15	-0.07	-0.22	-0.31	-0.25	-0.08
CLASP+	**+0.67**	**+0.42**	**+0.33**	**+0.49**	**+0.88**	**+1.16**	**+1.45**
CLASP	+0.23	+0.18	+0.16	+0.17	+0.40	+0.57	+0.84
CLASP w/o ICT	+0.52	+0.35	+0.27	+0.39	+0.75	+0.98	+1.27
CLASP w/o LACT	+0.23	+0.16	+0.14	+0.18	+0.35	+0.51	+0.74

Table 3: Improvements over base ST models and ablation studies. CLASP+ is the variant where the pseudo labels $\hat{y}_i$ is retrieved after each stage.

Vanilla contrastive self-training does not improve over the base model (except on LF-Wikipedia-500K), while CLASP does. As they both use pseudo labels predicted $\mathcal{E}$ before training, this indicates effectiveness of CLASP in controlling pseudo label noise.

Meanwhile, the effects of ICT and LACT depend on dataset type. For Amazon datasets, the overall best variant is CLASP+. However, for Wikipedia datasets, the best variant is CLASP with ICT removed. Further explaination in appendix G conclude that ICT (and their for the used of pseudo labels of the model after ICT in CLASP+) in fact encourages relying on lexical overlap and thus harm classification performance on Wikipedia data.

4 Conclusion

In this work we addressed ZXMTC, characterized by the absence of labeled data and an extremely large label space. We proposed a curriculum-guided contrastive self-training framework including label-aware cloze task as a pseudo-label-guided adversarial technique for unsupervised constrastive learning. Experiments demonstrate consistent improvements over standard self-training and obtain state-of-the-arts ZXMTC when combined with a compact pre-trained encoder.

[2](Zhu and Zamani, 2024) did not conduct experiments on very large label spaces such as LF-Wikipedia-500K and LF-Amazon-1M

[3]at step $t = T_1$ and step $t = T_1 + T_2$

5 Limitations

First, like most ZXMTC approaches, it strongly depends on meaningful label descriptions. Second, the method assumes a static label space and does not explicitly address dynamic or evolving labels. Finally, part of the observed performance gains comes from the underlying pre-trained encoder. Nevertheless, the proposed curriculum-guided contrastive framework consistently improves results beyond the base model alone, showing that it provides complementary benefits.

References

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.

Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit Dhillon. 2020. Taming pretrained transformers for extreme multi-label text classification.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep contrastive learning for unsupervised textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 879–895, Online. Association for Computational Linguistics.

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*, KDD '21, page 527–535, New York, NY, USA. Association for Computing Machinery.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2020. Revisiting self-training for neural sequence generation. In *International Conference on Learning Representations*.

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.

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.

Sujay Khandagale, Han Xiao, and Rohit Babbar. 2019. Bonsai – diverse and shallow trees for extreme multi-label classification.

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.

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*, WWW '18, page 993–1002, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Anand Rajaraman and Jeffrey David Ullman. 2012. *Mining of massive datasets*. Cambridge University Press, Cambridge.

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.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pages 109–126. National Institute of Standards and Technology (NIST).

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3:333–389.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers.

Qizhe Xie, Minh-Thang Luong, Eduard 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)*, pages 10684–10695.

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

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.

Hsiang-Fu Yu, Jiong Zhang, Wei-Cheng Chang, Jyun-Yu Jiang, Wei Li, and Cho-Jui Hsieh. 2022. Pecos: Prediction for enormous and correlated output spaces. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 4848–4849, New York, NY, USA. Association for Computing Machinery.

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.

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

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

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D. Cubuk, and Quoc V. Le. 2020. Rethinking pre-training and self-training.

A Evaluation Metrics for Information Retrieval

Let x_i be an instance of a dataset of n instances. Let $Y_i \subset Y$ be the set of relevant labels, and $\hat{Y}_i^k$ be the top-k labels based on similarity to the input query in the embedding space $\mathcal{E}$:

$$\hat{Y}_i^k = \underset{y \in Y}{\text{top}k} \left[\mathcal{E}(x_i) \cdot \mathcal{E}(y) \right] \tag{8}$$

P@k and R@k are calculated for each instance and averaged uniformly over all instances:

$$\text{P@}k = \frac{1}{n} \sum_{i=1}^{n} \frac{\left| \hat{Y}_i^k \cap Y \right|}{\left| \hat{Y}_i^k \right|} \tag{9}$$

$$\text{R@}k = \frac{1}{n} \sum_{i=1}^{n} \frac{\left| \hat{Y}_i^k \cap Y \right|}{|Y|} \tag{10}$$

B Details of baselines

- Lexical matching approaches:
  - **TF-IDF** (Rajaraman and Ullman, 2012) which represents instances and labels by sparse TF-IDF features and retrieve top labels based on similarity of these features.
  - **BM25** (Robertson et al., 1994) is a term-based ranking model that scores documents based on their term frequencies and document lengths.
  - **GloVe** (Pennington et al., 2014) uses dense average word embeddings trained on co-occurrence statistics to measure similarity between instances and labels.
- Contrastive representation learning:
  - **SentBERT** (Devlin et al., 2019; Reimers and Gurevych, 2019): a sentence encoder trained as siamese networks on semantic textual similarity task to derive semantically meaningful embeddings.
  - **SimCSE** (Gao et al., 2021) is an unsupervised contrastive learning method that constructs positive pairs by dropout as data augmentation, and fine-tunes a BERT siamese architecture.
  - **MPNet** (Song et al., 2020) is an another Sentence-BERT model originally designed for searching sentence paraphrases.
  - **ICT** (Lee et al., 2019) trains retriever models by selecting a sentence as query and the remaining of the document as context, then minimize the similarity of their embedding via a dual encoder architecture.
  - **TAS-B** (Hofstätter et al., 2021) trains a lightweight bi-encoder using supervision from a cross-encoder model and ColBERT on the MS MARCO dataset.

- **MACLR** (Xiong et al., 2022) utilizes a two-stage pre-training procedure. It leverages the ICT task to construct (context, title) pairs and employs unsupervised K-means clustering to generate additional positive pairs for contrastive learning.
- **RTS** (Zhang et al., 2022) focuses on the structural nature of documents. Breaking long content into random, non-overlapping contiguous subsequences and pairing them with titles to form positive pairs, it forces the model to capture the semantic similarity between different sections of a document and its central topic.

- LLM-based inference
  - **GPT3.5**: (Zhu and Zamani, 2024) provides GPT3.5 with solely the test input and the task objective, to generate a prediction, then adapt it to the actual label space.
  - **ICXML** (Zhu and Zamani, 2024) first generates candidate labels and then employs the LLM to rerank these candidates based on the test instance. It then utilize the in-context learning capabilities of LLM to generate prediction from reranked candidates.

C ZXMTC datasets

Table table 4 summarizes statistics of four studied datasets for ZXMTC.

Dataset	N_{train}	N_{test}	N_{label}
LF-Amazon-131K	294,805	134,835	131,073
LF-WikiSeeAlso-320K	693,082	177,515	312,330
LF-Wikipedia-500K	1,813,391	783,743	501,070
LF-Amazon-1M	914,179	1,465,767	960,106

Table 4: ZXMTC datasets: number of training (N_{train}) and test documents (N_{test}), number of labels (N_{label}).

D Base models statistics of baselines

Table 5 summarizes the number of parameters and the embedding dimension of base models used in diffent baselines.

Method	Base model	Parameters	Dimension	Sequence length
SentBERT	BERT-base	110M	768	512
SimCSE	BERT-base	110M	768	512
MPNet	MPNet-base	0.1B	768	512
ICT	BERT-base	110M	768	512
TAS-B	BERT-base	110M	768	512
MACLR	BERT-base	110M	768	512
RTS	MPNet-base	0.1B	768	512
GPT3.5	GPT3.5	175B		
ICXML	GPT3.5	175B		
Ours	All-miniLM-L12	33.4M	384	256

Table 5: Base model statistics of ZXMTC methods.

E Implementation Details

All experiments are implemented using the HuggingFace Transformers library and PyTorch. The backbone encoder is All-MiniLM-L12-v2 (appendix F).

During training, embeddings are optimized using the AdamW optimizer. The initial learning rate is set to 1×10^{-5} during pre-training, followed by linear decay.

In training, theoretically as large as possible for in-batch contrastive learning. Here training is performed on a H200 GPU with a batch size N of 128 due to memory limitation. The initial ICT pre-training stage is conducted for $T_1 = 25000$ steps. Curriculum-guided mixed LACT+ST stages last for $T_2 = 50000$ steps and finally 25000 steps of self-training.

F Benchmarking pre-trained semantic sentence transformers

In our preliminary experiments, we benchmark following sentence transformers. These are general purpose models that has been pre-trained for semantic similarity on a concatenated dataset of over 1 billion sentence pairs[4]: All-MPNet-base-v2, All-MiniLM-L12-v2, All-MiniLM-L6-v1, All-DistilRoBERTa-v1. See table 6 for complete results on four datasets.

Table 6 shows the performance of different pre-trained sentence transformers, without any further training on any individual dataset. Notice that despite having smaller size (table 5), compared to MPNet and DistilRoBERTa architectures, the MiniLM family offers strong performance, without any learning methods. We choose All-MiniLM-L12-v2 for following experiments as the base model for our training procedure due to its smaller size.

On the other hand, when compared with other baselines in the following section, the sentence transformers in study also show great performance. We attribute this achievement to their pre-training,

[4]For more details, model cards are available at https://www.sbert.net/docs/sentence_transformer/pretrained_models.html

Base model	P@1	P@3	P@5	R@1	R@3	R@5	R@10	R@100
LF-Amazon-131K								
All-MPNet-base-v2	19.48	12.77	9.63	11.23	20.36	24.83	30.65	**48.57**
All-MiniLM-L12-v2	**20.45**	**13.67**	**10.21**	**11.65**	**21.58**	**26.89**	**31.66**	47.78
All-MiniLM-L6-v2	18.56	12.44	9.37	10.59	19.73	24.03	29.62	34.70
All-DistilRoBERTa-v1	18.80	12.39	9.37	10.69	19.71	24.05	29.44	45.35
LF-WikiSeeAlso-320K								
All-MPNet-base-v2	**23.96**	**15.72**	**12.10**	**13.20**	**23.16**	**28.16**	**34.95**	**56.76**
All-MiniLM-L12-v2	23.43	15.34	11.76	12.81	22.41	27.15	33.58	53.96
All-MiniLM-L6-v2	21.92	14.50	11.15	11.90	21.14	25.74	32.09	52.95
All-DistilRoBERTa-v1	19.55	13.08	10.15	10.42	18.79	23.15	29.28	50.74
LF-Wikipedia-500K								
All-MPNet-base-v2	27.13	16.74	12.58	9.50	16.62	20.21	25.39	44.70
All-MiniLM-L12-v2	**30.27**	**18.05**	**13.40**	**10.78**	**18.16**	**21.84**	**27.05**	**45.41**
All-MiniLM-L6-v2	24.96	15.24	11.48	8.95	15.47	18.87	23.82	42.30
All-DistilRoBERTa-v1	10.87	7.06	5.51	4.05	7.46	9.44	12.61	27.86
LF-Amazon-1M								
All-MPNet-base-v2	15.63	9.66	7.07	11.35	20.17	24.38	29.95	47.56
All-MiniLM-L12-v2	**17.13**	**10.60**	**7.72**	**12.47**	**22.13**	**26.61**	**32.40**	**49.99**
All-MiniLM-L6-v2	16.50	10.22	7.47	12.01	21.34	25.73	31.42	48.93
All-DistilRoBERTa-v1	16.13	9.91	7.22	11.71	20.66	24.86	30.27	46.77

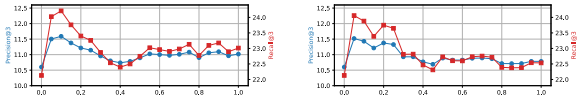
Table 6: Evaluation in ZSL of base sentence transformers for ZXMTC.

where they learn to produce semantically rich embeddings. The pre-training effectiveness is a result of the large mix dataset constructed from various sources: Q&A (Quora, PAQ, Stack Exchange, MS MARCO), research citations (S2ORC), NLI (SNLI, MultiNLI), paraphrasing (WikiAnswers, Stack Exchange), etc. Even though not trained on the training dataset (in an unsupervised manner like TF-IDF, SimCSE, MACLR, RTS), the large mix corpus allows sentence transformers to learn general semantic, covering the in-domain distribution of the datasets.

G On the effect of ICT with respect to data type

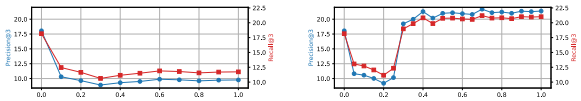
In this section, we first study the difference in performance between CLASP and CLASP+ versions of our algorithm, then the influence of ICT and LACT to training for different types of dataset.

We observe that titles in Amazon datasets often partially resemble the correct labels. For Amazon datasets, document title is similar to one of the correct labels, thus ICT is effective. In figure 1, both CLASP and CLASP+ variants show a performance gain in the ICT in the first $T_1 = 25\%T$ steps. At the end of this stage, the performance drops as the ICT deviates from classification, indicating that



(a) CLASP+ (b) CLASP

Figure 1: Evolution of P@3 and R@3 with respect to t/T, on LF-Amazon-1M



(a) CLASP+ (b) CLASP

Figure 2: Evolution of P@3 and R@3 with respect to t/T, on LF-Wikipedia-500K

ICT may initially enhances in-domain representations but later is not coherent with the classification task.

Meanwhile, titles in Wikipedia datasets are short terms and often repeated in the content texts. As the document body already contains its title, the model tends to rely on lexical cues as a learning shortcut. This trick may harm classification, shown by a remarkable decline in metrics in the first $T_1 = 25\%T$ steps shown in figure 2. Reusing $\mathcal{E}(0)$ to retrieve pseudo labels actually helps to regain performance from the degraded model $\mathcal{E}(t = T_1)$ (figure 2b),

showing the improvements made by our adversarial proposition. Self-training $\mathcal{E}(t = T_1)$ from its prediction never recover the initial performance (figure 2a).

Same as observations made in section 3.3, we notice that the effects of ICT and LACT depends on dataset type. We confirm the previous explanation. Wikipedia datasets contain documents where titles are repeated in the content texts. In this case, ICT objective fallbacks to term overlapping as a learning shortcut. ICT thus generates noise and becomes harmful to training. At the same time, the effectiveness of LACT is validated, showing the best performance gain.