Leveraging Pre-Trained Extreme Multi-Label Classifiers for Zero-Shot Learning

Natalia Ostapuk
University of Fribourg
Fribourg, Switzerland
natalia.ostapuk@unifr.ch

Ljiljana Dolamic
armasuisse
Switzerland
ljiljana.dolamic@ar.admin.ch

Alain Mermoud
armasuisse
Switzerland
alain.mermoud@armasuisse.ch

Philippe Cudré-Mauroux
University of Fribourg
Fribourg, Switzerland
pcm@unifr.ch

***Abstract*—Extreme multi-label (XML) classification involves assigning multiple labels to an instance from an extremely large set of possible labels. Despite its significance, *zero-shot learning* within the context of XML classification remains relatively understudied. Zero-shot learning becomes pivotal when dealing with new labels not present during the training phase, a common occurrence in real-world applications. Existing approaches often resort to training zero-shot learning classifiers from scratch, which can be computationally expensive and may not fully exploit the knowledge embedded in pre-trained models. In this paper, we propose a novel approach to address this gap by introducing a method for transferring knowledge from a pre-trained XML classifier to enhance zero-shot learning capabilities. We present experimental results that demonstrate the potential of knowledge transfer from pre-trained XML classifiers as a promising avenue for advancing zero-shot learning in the challenging context of extreme multi-label classification.**

***Index Terms*—zero-shot learning, extreme multi-label classification, semantic embeddings**

I. Introduction

Extreme multi-label (XML) text classification is the task of assigning the most relevant labels taken from an extremely large label set to a given document. One of the applications of XML classification is to represent the semantic content of a document with its key concepts (semantic text tagging). This task is of particular importance for scientific document collections: as the number of scientific papers getting published is rapidly increasing, semantic tagging becomes crucial to support the discovery of new scientific results as well as exploratory efforts within a new field of interest.

This paper addresses a zero-shot learning scenario of XML, i.e., predicting *unseen*, or zero-shot labels, which are not present during the training phase. Zero-shot labels frequently occur in XML tasks, as label sets evolve over time and new labels arise. To incorporate new labels, traditional classification approaches would require to relabel the training set and to retrain the model from scratch. This process is particularly time-consuming and demanding when dealing with extreme scales, where the size of the training set can reach millions of data points [1]. The majority of XML classifiers circumvent this problem and operate under the assumption that the label space is fixed, and thus are inherently incapable of predicting unseen labels [2]–[4].

The common approach to zero-shot learning classification is to project data points and labels close together in a dense shared embedding space and subsequently leverage nearest neighbor search to predict labels relevant to a given data point. A new label is then included into the scheme by mapping it onto the same space, enabling it to be handled alongside existing labels. Although this approach proved effective [5], [6], it still requires to devise and train complex models to learn effective embeddings and representations of the input features and labels. On the other hand, there exist a handful of powerful XML classifiers that have been pre-trained and optimized for specific datasets and that showcase high performance in their specialized domains. Intuitively, knowledge learned from a traditional XML task can be reused to boost the performance on zero-shot label prediction. As such, the research question we aim to explore in this paper is the following:

RQ: Is it viable to transfer knowledge from a pre-trained XML classifier to enhance zero-shot label prediction?

More specifically, we selectively combine semantic embeddings of seen labels predicted by a pre-trained XML classifier weighted by their corresponding probabilities. Subsequently, we compare the resulting representation with previously unseen labels, selecting the most similar candidates. For example, if a classifier predicts labels Convolution and Artificial neural network with probabilities 0.96 and 0.81 for a given document, their combined representation might be similar to the representation of a new label Convolutional Neural Network:

$$0.96 * emb_{Conv.} + 0.81 * emb_{ANN} \approx emb_{CNN}$$

To further enhance prediction accuracy, we combine the representation obtained from predicted labels with the input document representation.

Our method for zero-shot label prediction offers numerous advantages compared to traditional approaches:

1) **Efficiency:** it does not require additional training and can be employed as an add-on to existing XML models.
2) **Compatibility:** it is compatible with any probabilistic XML classifier, allowing it to fully benefit from recent advances in the field of XML classification.
3) **Scalability:** it is agnostic to the size of the label set, making it adaptable to large label sets and allowing for a scalable implementation.

II. Related Work

Extreme Multi-Label Classification. Traditional XML approaches can be divided into three groups [7]: (i) *one-vs-all* methods independently train a binary classifier for each label [8], [9], (ii) *tree-based* methods recursively partition the instance set or the label set and at each non-leaf node train a classifier focusing on a small subset of the original problem [10], while (iii) *embedding-based* methods project labels onto a low-dimensional vector space and perform classification using nearest-neighbor search [11], [12].

Over the last few years, there have been a growing number of works that demonstrate the efficiency of *deep learning* for XML classification [2]–[4]. Among these, AttentionXML [2] is notable for its high prediction accuracy and consistent performance across various datasets. AttentionXML features a BiLSTM architecture and leverages the attention mechanism to capture fine-grained dependencies between input documents and labels. We use AttentionXML as a base XML classifier in the present work.

Zero-Shot Learning. While zero-shot classification has been extensively studied, the majority of existing research primarily focuses on *multi-class* learning, with only a limited number of approaches addressing *multi-label* learning. Among them, ZestXML [5] achieves state-of-the-art performance on the Generalized Zero-Shot XML task through projecting data points and labels onto the same high-dimensional vector space. Rios et al. [13] leverage Graph Neural Networks for incorporating structural information about the label space into label representation. Mullenbach et al. [14] utilize a convolutional network with attention for predicting medical codes from clinical texts. Chalkidis et al. [6] conducted a notable study, empirically evaluating various XML methods in the context of zero-shot learning. Among multi-class approaches, ConSE [15] stands out as a noteworthy example that inspired our present work. ConSE predicts zero-shot classes by employing a convex combination of semantic embeddings derived from classes that were available during training. How to adapt this approach to multi-label classification, however, remains an open research question.

III. Method

Our approach, hereafter referred as XML0, consists of two components: a label fusion module and a semantic similarity module. We first generate a combined representation of *seen* labels predicted by a pre-trained XML classifier and calculate the cosine similarity between this representation and *unseen* labels (label fusion module). Next, we measure the cosine similarity between the input document and unseen labels (semantic similarity module). Finally we combine both scores for each unseen label to produce the final ranking. The overall architecture of XML0 is depicted in Figure 1. In the following, we describe each of those components in more detail.

A. Label Fusion Module

The objective of label fusion is to leverage knowledge obtained by a pre-trained XML classifier for predicting new labels that were not part of the initial training dataset. Our label fusion module was motivated by ConSE [15], which maps images onto the semantic embedding space via a convex combination of the class embedding vectors. For example, if an image is classified as $lion$ with probability 0.6 and as $tiger$ with probability 0.4, then the predicted semantic embedding, $f(x) = 0.6 \cdot s(lion) + 0.4 \cdot s(tiger)$, will point to somewhere between $lion$ and $tiger$ in the semantic space, which might be close to a new, previously unseen class $liger$.

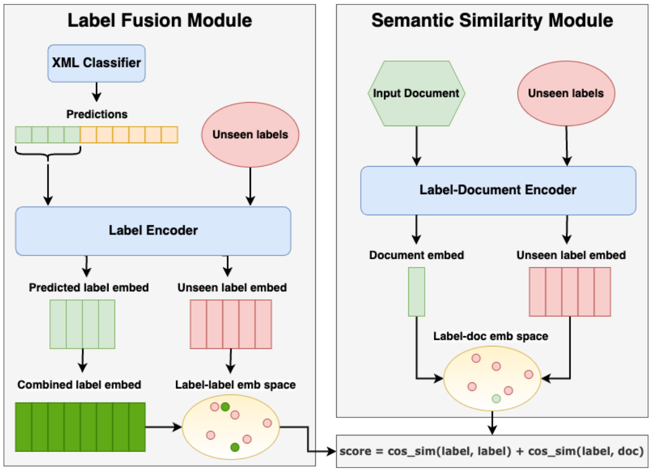

Fig. 1. XML0 Architecture - The diagram illustrates the key components of XML0: the label fusion module (left), the semantic similarity module (top right), and the final score generation module (bottom right).

However, extending this approach to the multi-label classification problem is not straightforward. In contrast to *multi-class* classification, in a *multi-label* scenario each instance can belong to multiple classes, and their convex combination does not necessarily produce a meaningful new class. For example, the following labels, assigned to a single document, encompass at least three distinct topics (highlighted with different colors) discussed within the text: *[Artificial intelligence, Medical imaging, Computer vision, Modular design, Pipeline (software)]*. A new label can potentially result from any combination of these labels or diverge entirely from them. Building on this observation, our label fusion module performs an exhaustive search among all possible combinations of predicted labels selecting the most meaningful combination w.r.t. each new label.

Our label fusion module requires a pre-trained XML classifier $\mathcal{X}$ and a label encoder $\mathcal{E}$ as input. The classifier estimates probabilities of a given document d belonging to each class (label), while the encoder generates semantic embeddings for labels from their textual form (title and/or definition). The label fusion algorithm starts with applying an XML classifier to a given document, obtaining estimated probabilities for *seen* labels. Following this, it identifies a subset of these labels, denoted as P, by selecting labels with predicted probabilities surpassing a specified threshold, and generates all possible combinations $\{C_n^k | 1 \leqslant k \leqslant n\}$ from this selected subset. Subsequently, for every combination $c_j \in C_n^k$, a unified semantic representation is generated as the sum of the semantic embeddings of labels weighted by their corresponding probabilities:

$$repr(c_j) = \sum prob(l_i) \cdot \mathcal{E}(l_i),$$

where $l_i \in c_j$ and $prob(l_i) = \mathcal{X}(l_i|d)$. In the final step, the algorithm computes the cosine similarity between combinations and unseen labels. For each label, it chooses the combination with the highest similarity score. Subsequently, unseen labels are ranked based on these scores. The above process is described by the high-level pseudo code in Algorithm 1.

Algorithm 1 Label fusion algorithm

```
Input: XML classifier $\mathcal{X}$, label encoder $\mathcal{E}$, threshold $t$, document $d$, unseen labels $U = \{l^u\}$, seen labels $S = \{l^s\}$
Output: ranked unseen labels $U_{ranked}$
$O \leftarrow$ evaluate $\mathcal{X}(d, S)$;
$P \leftarrow \{(l_i^s, prob_i) \,|l_i^s \in S, prob_i \in O, prob_i > t\}$;
$C_n^k \leftarrow$ generate all possible combinations from $P$;
$C^{emb} \leftarrow \emptyset$;
for $c_j \in C_n^k$ do
    $c_j^{emb} \leftarrow 0$;
    for $(l_i^s, prob_i) \in c_j$ do
        $c_j^{emb} += prob_i \cdot \mathcal{E}(l_i^s)$;
    end for
    add $c_j^{emb}$ to $C^{emb}$;
end for
for $l_i^u \in U$ do
    $rank_i \leftarrow max(cos\_sim(C^{emb}, l_i^u))$;
end for
$U_{ranked} \leftarrow$ rank $l_i^u \in U$ by $rank_i$;
return $U_{ranked}$;
```

B. Semantic Similarity Module

The goal of the semantic similarity module is to integrate the content of an input document to further boost the performance of XML0 on zero-shot labels.

During this stage, we encode both the input documents and the textual representations of unseen labels, subsequently measuring the pairwise cosine similarity between them. Following this computation, labels are systematically ranked based on their similarity scores with respect to each document. This ranking process provides a structured evaluation of label relevance and alignment with the content of the input documents.

Depending on the dataset under consideration, the textual representation of a label can be articulated through its title and/or definition. In our experimental framework, we adopt the title and the initial paragraph of the abstract extracted from the corresponding Wikipedia page as the means to represent the labels.

Following recent advances in language modeling using attention-based transformers, we adopt Sentence-BERT (SBERT) [16] as an encoder for documents and labels. SBERT is a modification of the BERT model [17] able to derive fixed-sized vectors for input sentences. To refine SBERT for our specific task, we engaged in further fine-tuning. Specifically, we randomly selected two million (d, l) pairs from the training set, where d represents a document and l denotes the textual form of a label. During the training process, our objective was to minimize the cosine similarity between d and l if l is relevant to d and to maximize it otherwise. This fine-tuning strategy aimed to enhance SBERT's capacity to capture and distinguish the nuanced semantic relationships between documents and their associated labels.

TABLE I
DATASET STATISTICS

	N_{train}	N_{test}	L_{total}	L_{zs}	$\overline{W_d}$	$\overline{W_l}$
MAG-CS	560,058	18,024	15,303	504	87	73.8

N_{train}: #training instances, N_{test}: #test instances, L_{total}: #labels total, L_{zs}: #labels zero-shot, $\overline{W_d}$: avg #words per doc, $\overline{W_l}$: avg #words per label.

TABLE II
PERFORMANCE COMPARISON OF XML0 AND OTHER COMPETING METHOD ON MAG-CS DATASET

Algorithms	Prec@1	Prec@3	Prec@5	nDCG@3	nDCG@5
ZESTXML	0.2275	0.1141	0.0760	0.2623	0.2751
CONSE	0.1943	0.1399	0.1040	0.3025	0.3406
SBERT	0.2800	0.1584	0.1131	0.3550	0.3871
XML0$_{LF}$	0.2478	0.1398	0.0996	0.3242	0.3537
XML0	**0.2994**	**0.1695**	**0.1223**	**0.3768**	**0.4143**

IV. EXPERIMENTS AND RESULTS

A. Experimental Setting

Dataset. In our work, we specifically focus on collections of scientific papers. Therefore, we evaluate our method using the Microsoft Academic Graph (MAG) [18] dataset. MAG is a heterogeneous graph containing scientific publication records labeled with relevant concepts. Following [19], for our experiments we use a subset of MAG focusing on the computer science domain - we refer to this dataset as MAG-CS. As MAG-CS does not contain unseen labels, we further generate a zero-shot version of it, randomly dropping 500 labels from the training set, while retaining them in the test set. To ensure the robustness of our evaluation, we repeated this process three times. In the rest of this section, we will report average values across the three zero-shot snapshots of MAG-CS. In this study, our main focus is on zero-shot labels. Consequently, we exclude labels that were part of the training set from the test set, aiming to assess the model's performance specifically on unseen labels. We report important statistics from our dataset in Table I.

Baselines. We compare our method against the following zero-shot classification approaches:

- **ZestXML** [5] focuses on the Generalized Zero-shot XML (GZXML) task, whose goal is to select relevant labels from both seen and unseen labels. ZestXML learns to project a data point's features close to the features of its relevant labels through a highly sparsified linear transform. To ensure a fair comparison, we assess the performance of ZestXML exclusively on unseen labels.
- **ConSE** [15] is an image classification model, which maps images into the semantic embedding space via a

convex combination of the embedding vectors of classes, predicted for each image. To apply ConSE to our task, we implemented a naive version of our label fusion module, which considers a single combination of top predicted labels, i.e., C_n^n.

- **SBERT** [16] is a sentence embedding model that utilizes a siamese network architecture to learn semantically meaningful representations for sentences in a way that preserves their pairwise semantic similarity. In the context of the present work, the SBERT baseline corresponds to our semantic similarity module without label fusion.

We additionally evaluated the ablated version of our model, XML0_{LF}, which includes the Label Fusion module only, to understand the contribution of this specific module to the overall performance. In both XML0 and XML0_{LF} we used AttentionXML [2] as the XML classifier and SBERT with `all-MiniLM-L6-v2`[1] as the label encoder.

Metrics. In line with previous XMLC works [2], [4], [19], we use $P@k$ (Precision at k) and $nDCG@k$ (normalized Discounted Cumulative Gain at k) as evaluation metrics. $P@k$ is defined as the number of correct predictions considering only the top k elements divided by k. Discounted cumulative gain (DCG) measures the quality of rankings, assigning higher scores to hits at top ranks. nDCG is a normalized version of DCG, which accounts for the varying number of positive labels per instance.

B. Results and Discussion

We summarize the results of our experiments in Table II. Our proposed approach outperforms all competing methods, which demonstrates the effectiveness of integrating information obtained from input documents with knowledge transferred from a pre-trained XML classifier.

The comparison with the SBERT model is particularly intriguing, as it also serves as an ablation study. The SBERT baseline corresponds to the semantic similarity module of our approach, and highlights the performance improvement achieved by incorporating information from predicted labels.

The observation that XML0_{LF} outperforms ConSE on most metrics validates our hypothesis that, in multi-label classification, a straightforward combination of all top predicted labels may lead to sub-optimal performance. This supports our decision to conduct a comprehensive search across combinations.

Surprisingly, ZestXML exhibits the lowest performance, despite the promising results reported by [5] on other datasets. This phenomenon might in part be attributed to the specific characteristics of the MAG-CS dataset. Indeed, the labels in MAG-CS boast extensive textual descriptions, being linked to Wikipedia articles. ZestXML employs Bag-of-Words (BoW) feature vectors for label encoding, while other methods leverage transformers. Transformers excel at capturing sequential information and handling long-term dependencies, making them more suitable for understanding the complexities of longer texts compared to traditional BoW models.

[1] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

V. Conclusion and Future Work

In this paper, we explored the viability of leveraging labels predicted by a pre-trained classifier for the task of zero-shot XML classification. Our experimental results demonstrate the potential of this approach and suggest a valuable avenue for enhancing the efficiency and adaptability of zero-shot XML classifiers. As part of future work, we aim to devise a more efficient method for retrieving the optimal combination of labels and validate our results on additional datasets.

References

[1] J. J. McAuley and J. Leskovec, “Hidden factors and hidden topics: understanding rating dimensions with review text,” in *7th ACM Conf. on Recommender Syst., RecSys ’13*. ACM, 2013, pp. 165–172.

[2] R. You *et al.*, “Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification,” in *Advances in Neural Info. Process. Syst. 32, NeurIPS 2019*, 2019.

[3] J. Zhang, W. Chang, H. Yu, and I. S. Dhillon, “Fast multi-resolution transformer fine-tuning for extreme multi-label text classification,” in *Advances in Neural Info. Process. Syst. 34, NeurIPS 2021*, 2021.

[4] J. Liu, W. Chang, Y. Wu, and Y. Yang, “Deep learning for extreme multi-label text classification,” in *Proc. of 40th ACM SIGIR Conf. on Res. and Develop. in Inf. Retrieval*. ACM, 2017, pp. 115–124.

[5] N. Gupta, S. Bohra, Y. Prabhu, S. Purohit, and M. Varma, “Generalized zero-shot extreme multi-label learning,” in *KDD ’21: 27th ACM SIGKDD Conf. on Knowl. Discovery and Data Mining*. ACM, 2021.

[6] I. Chalkidis *et al.*, “An empirical study on large-scale multi-label text classification including few and zero-shot labels,” in *Proc. of 2020 Conf. on Empirical Methods in Natural Lang. Process., EMNLP 2020*. Association for Computational Linguistics, 2020, pp. 7503–7515.

[7] W. Liu, X. Shen, H. Wang, and I. W. Tsang, “The emerging trends of multi-label learning,” *CoRR*, vol. abs/2011.11197, 2020.

[8] I. E. Yen, X. Huang, P. Ravikumar, K. Zhong, and I. S. Dhillon, “Pd-sparse : A primal and dual sparse approach to extreme multiclass and multilabel classification,” in *Proc. of 33d Int. Conf. on Mach. Learn., ICML 2016*, 2016, pp. 3069–3077.

[9] R. Babbar and B. Schölkopf, “Dismec: Distributed sparse machines for extreme multi-label classification,” in *Proceedings of 10th ACM Int. Conf. on Web Search and Data Mining, WSDM 2017*, 2017.

[10] Y. Prabhu and M. Varma, “Fastxml: a fast, accurate and stable tree-classifier for extreme multi-label learning,” in *20th ACM SIGKDD Int. Conf. on Knowl. Discovery and Data Mining, KDD ’14*. ACM, 2014.

[11] Y. Tagami, “Annexml: Approximate nearest neighbor search for extreme multi-label classification,” in *Proc. of 23rd ACM SIGKDD Int. Conf. on Knowl. Discovery and Data Mining*. ACM, 2017, pp. 455–464.

[12] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain, “Sparse local embeddings for extreme multi-label classification,” in *Advances in Neural Info. Process. Syst. 28, NeurIPS 2015*, 2015, pp. 730–738.

[13] A. Rios and R. Kavuluru, “Few-shot and zero-shot multi-label learning for structured label spaces,” in *Proc. of 2018 Conf. on Empirical Methods in Natural Lang. Process.* Association for Computational Linguistics, 2018, pp. 3132–3142.

[14] J. Mullenbach, S. Wiegreffe, J. Duke, J. Sun, and J. Eisenstein, “Explainable prediction of medical codes from clinical text,” *CoRR*, vol. abs/1802.05695, 2018.

[15] M. Norouzi *et al.*, “Zero-shot learning by convex combination of semantic embeddings,” in *2nd Int. Conf. on Learn. Representations, ICLR 2014, Conf. Track Proc.*, 2014.

[16] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” in *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, 2019, pp. 3980–3990.

[17] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” *CoRR*, vol. abs/1810.04805, 2018.

[18] K. Wang *et al.*, “Microsoft academic graph: When experts are not enough,” *Quant. Sci. Stud.*, vol. 1, no. 1, pp. 396–413, 2020.

[19] Y. Zhang, Z. Shen, Y. Dong, K. Wang, and J. Han, “MATCH: metadata-aware text classification in A large hierarchy,” in *WWW ’21: The Web Conference 2021*. ACM / IW3C2, 2021, pp. 3246–3257.