RESEARCH-ARTICLE

# Extreme Multi-Label Completion for Semantic Document Tagging with Taxonomy-Aware Parallel Learning

**JULIEN AUDIFFREN**, University of Fribourg, Fribourg, FR, Switzerland

**CHRISTOPHE BROILLET**, University of Fribourg, Fribourg, FR, Switzerland

**LJILJANA DOLAMIC**, Armasuisse, Switzerland, Bern, BE, Switzerland

**PHILIPPE CUDRE-MAUROUX**, University of Fribourg, Fribourg, FR, Switzerland

**Open Access Support** provided by:

**University of Fribourg**

**Armasuisse, Switzerland**



**Published:** 10 November 2025

**Citation in BibTeX format**

CIKM '25: The 34th ACM International Conference on Information and Knowledge Management
*November 10 - 14, 2025*
*Seoul, Republic of Korea*

**Conference Sponsors:**
SIGWEB
SIGIR

# Extreme Multi-Label Completion for Semantic Document Tagging with Taxonomy-Aware Parallel Learning

Julien Audiffren
Department of Computer Science
University of Fribourg
Fribourg, Switzerland
julien.audiffren@unifr.ch

Christophe Broillet
Department of Computer Science
University of Fribourg
Fribourg, Switzerland
christophe.broillet@unifr.ch

Ljiljana Dolamic
armasuisse S&T
armasuisse
Thun, Switzerland
ljiljana.dolamic@ar.admin.ch

Philippe Cudré-Mauroux
Department of Computer Science
University of Fribourg
Fribourg, Switzerland
philippe.cudre-mauroux@unifr.ch

## Abstract

The objective of Extreme Multi-Label Completion (XMLCo) is to predict missing document labels drawn from a very large collection. Together with Extreme Multi-Label Classification (XMLC), XMLCo is arguably one of the most challenging document classification tasks, as the number of potential labels is generally very large compared to the number of labeled documents. The collection of labels is often structured in a taxonomy that encodes relationships between labels, and many methods have been proposed to leverage this hierarchy to improve XMLCo algorithms. In this paper, we propose a new approach to this problem: TAMLEC (Taxonomy-Aware Multi-task Learning for Extreme multi-label Completion)[1]. TAMLEC divides the problem into several Taxonomy-Aware Tasks, i.e. into specific subsets of the labels drawn from paths in the taxonomy, and trains on these tasks using a dynamic Parallel Feature sharing approach where parts of the model are shared between tasks while others are task-specific. Then, at inference time, TAMLEC uses the labels available in a document to predict missing labels, using the Weak-Semilattice structure that is naturally induced by the tasks. Our empirical evaluation on real-world datasets shows that TAMLEC substantially outperforms the state of the art in XMLCo. Furthermore, additional experiments show that TAMLEC is particularly suited for few-shot settings, where new tasks or labels are introduced with only few examples after initial training.

## CCS Concepts

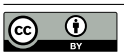
• **Applied computing** → **Document metadata**; • **Computing methodologies** → **Machine learning algorithms**; **Neural networks**; **Supervised learning by classification**.

## Keywords

Extreme Multi Label Completion, Transformers, Document Tagging, Taxonomies, Parallel Feature Sharing

**ACM Reference Format:**
Julien Audiffren, Christophe Broillet, Ljiljana Dolamic, and Philippe Cudré-Mauroux. 2025. Extreme Multi-Label Completion for Semantic Document Tagging with Taxonomy-Aware Parallel Learning. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3746252.3761233

[1] https://github.com/Jythen/Tamlec




*CIKM '25, Seoul, Republic of Korea*

ACM ISBN 979-8-4007-2040-6/2025/11
https://doi.org/10.1145/3746252.3761233

## 1 Introduction

In the past decades, the number of textual documents (documents for short hereafter) available online has increased dramatically [13]. This rise of large document collections has been further amplified in the last few years by Large Language Models and their ever-increasing need for new data corpora [40]. As a result, the automated labeling of documents has become a crucial issue [41], as labels allow to categorize documents, but also help with information search and enable users to easily navigate vast collections of documents [38]. The problem of assigning to each document a subset of labels drawn from a large collection of labels is referred to as Extreme Multi-Label Classification (XMLC) [41]. In the case where some documents are already equipped with an incomplete set of labels, the prediction of the missing labels is called Extreme Multi-Label Completion (XMLCo) [26]. Compared to XMLC, XMLCo methods are able to leverage known labels (whenever available) to infer additional information about the documents and improve their predictions. The problem of incomplete labels is common in many applications [31], and can originate from time constraints, subjectivity of the annotators, or the introduction of new labels over time. As such, label completion is key in improving the quality of existing datasets [30]. These problems are arguably some of the most challenging document classification tasks [22], for two main reasons. First, the set of labels to choose from is typically very large, often reaching tens of thousands, and each document may possess an arbitrary number of labels (the *labels scale* problem). Second, the data is often considered sparse, as many labels only have few training instances, and the problem is further compounded when considering collections of labels (the *scarcity* problem). Put together,

these issues make the use of traditional classification algorithms difficult and has given rise to new approaches.

Some of the most successful methods proposed in the past few years stem from the use of a label taxonomy [2, 26, 43]. Indeed, many real-world Extreme Multi-Label (XML) problems come equipped with a hierarchical label taxonomy, which is developed to facilitate the management of large collections of labels. These hierarchies encode relationships between labels that arise from their real-world usage, for instance through subsumption hierarchies specifying that a label (e.g. Computer Science) defines a more general class than a second label (e.g. Machine Learning). As a result, these structures yield valuable information for XML tasks, and in particular for XMLCo, as it has been observed that the documents that are only partially labeled are typically equipped with general, high-level labels, while more specific labels are often missing [38]. Notable XML-related taxonomies include the MeSH thesaurus[2] and the Microsoft Academic Graph (MAG) [38]. Previous work has shown that these taxonomies can be used to improve XML methods in multiple ways [16, 26, 43]. However, these works have focused on using the taxonomy to address the label scale problem only, and we argue that additional performance can be gained by further leveraging taxonomies to also alleviate the scarcity problem.

Multi-Task Learning (MTL) is a Machine Learning paradigm that involves training a model on multiple tasks simultaneously, with the goal of improving the performance of each task [3]. Data scarcity stands at the heart of the MTL challenge, as each task contains limited data, and training independent models on each task may result in poor performance and overfitting. One of the most popular MTL approaches is Parallel Feature Sharing, where multiple tasks are trained simultaneously by sharing a common feature extractor while maintaining task-specific components [45]. MTL and Parallel Feature Sharing have seen many applications in Natural Language Processing (NLP) and document analysis, and have received increased interest since the popularization of deep learning methods [10]. The idea behind Parallel Feature Sharing can be found in many deep-learning XML algorithms such as AttentionXML [41]. Indeed, in most architectures the documents are processed through a shared neural network architecture, and only differentiated in the final result, which is generally a vector containing the predicted relevance of each label for the input document. In that regard, only the weights of the last layers may be considered partially label-specific. However, and to the best of our knowledge, the use of more advanced MTL methods and in particular more intertwined shared/task-specific architectures and training methods have not received the same level of attention.

In this paper, we show that adapting the Parallel Feature Sharing paradigm to the taxonomy structure of the labels leads to further improvements. More precisely, we introduce TAMLEC (Taxonomy-Aware Multitask Learning for Extreme multi-label Completion), which uses ideas from XMLCo and MTL to better combine information sharing and the label taxonomy. To achieve this, TAMLEC first creates Taxonomy-Aware Tasks (TATs) on subsets of the labels that are adapted to the taxonomy structure and to semantic constraints (see Section 3.1). TAMLEC uses a modified transformer architecture that is adjusted to the characteristics of the TATs by balancing shared neurons and task-specific neurons to improve prediction performance. To predict missing labels, TAMLEC uses the known labels of each document to choose their relevant TATs, and predict paths of labels on each of the selected task, which are then combined to perform the final prediction. Our contributions can be summarized as follows:

- We extend previous work on tree-based taxonomy (such as [26]) to accommodate a more general structure, Weak Semilattices, that naturally arises in many XML problems.
- We introduce the Taxonomy-Aware Tasks (TATs) decomposition, where the taxonomy is decomposed into subtasks that satisfy coherence and separability criteria.
- We propose a new XMLCO algorithm, TAMLEC, that leverages an adaptive transformer architecture with a new TATs-aware loss to balance information sharing among tasks.
- We perform an extensive empirical evaluation of our method, and show that it outperforms the state of the art on XMLCo problems. Furthermore, our results show that TAMLEC is particularly suited for handling XML few-shot tasks, where the new labels that form the TATs are introduced after training on a few examples only.

The rest of the paper is organized as follows. Section 2 summarizes the related work in XML and MTL. We present our different contributions (TAMLEC, TATs, etc.) in Section 3. Finally, the experimental evaluation of TAMLEC is presented in Section 4.

## 2 Background and Related Work

***Extreme Multi-Label (XML).*** Several strategies have been proposed to address the challenges of XML [23]. These works include XML-CNN [21], AttentionXML [41], as well as X-Transformer [8] (see Section 4.1 for additional details on these methods). In parallel, there has been significant interest in the use of the structure organizing the labels, such as taxonomies, to enhance XML methods [15, 16]. Notably, MATCH [43] used a transformer architecture while leveraging the label structure through the use of regularization, by enforcing each label to be similar to its parents, while [2] introduced a recurrent neural network, whose predictions are combined across levels of the taxonomy to improve results. Arguably the model closest to us is Hector [26], where the authors proposed an XMLCo algorithm that leverages the taxonomy tree by using a transformer architecture to directly predict a path on the label structure, using known labels to form the path prefix. However, while we also use a modified transformer architecture to predict paths on a label structure, compared to their work, we extend the framework to a much more general structure, Weak-Semilattice, that better captures the subtleties of taxonomies, which we combine with a completely different approach based on Parallel Feature Sharing and Taxonomy Aware Tasks (TATs). Our experiments show that our method, TAMLEC, outperforms existing XMLCo algorithms.

***Multi-Task Learning (MTL).*** The MTL paradigm has encountered significant success since the seminal work of [3]. Many methods have been developed around leveraging multiple dependent and smaller tasks to improve the performance of the resulting larger model [44, 45], and examples of successful MTL applications include cancer detection [11, 12], web search ranking [9], and many NLP tasks [1, 7, 10]. Feature sharing MTL is arguably one of the

[2] https://www.nlm.nih.gov/mesh/meshhome.html

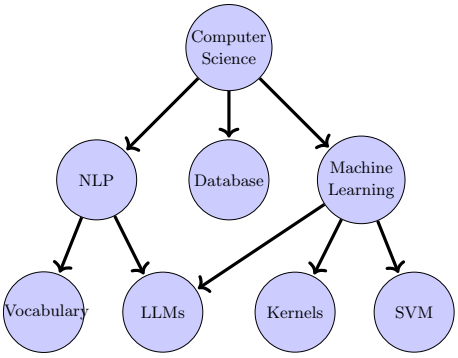



**Figure 1: Toy scientific taxonomy. An arrow from $\ell_1$ to $\ell_2$ represents $\ell_1 \leq \ell_2$. This taxonomy can be represented with a Weak-Semilattice, but not with a tree – as "LLMs" has multiple parents.**

most popular MTL approaches in deep learning architectures, and while TAMLEC approach takes inspiration from it, to the best of our knowledge, TAMLEC is the first to intertwine MTL into the taxonomy structure for XMLCo. Furthermore, one of our main contributions is a new taxonomy-adapted loss (Section 3), and our results (see Section 4) show that it indeed represents a very promising avenue for solving taxonomy-based XMLC problems.

***XML Few-Shot Classification.*** In this work, we also consider the XML few-shot classification problem, where models are first primed with a large dataset, before being trained to recognize new labels with only a limited number of training examples [33]. This setting reflects the common situation where new labels – such as emerging research areas (e.g. "LLMs") – are introduced over time into taxonomies. As retraining XMLC algorithms from scratch on the new taxonomy is particularly costly, several alternatives have been proposed. Some of the most influential methods in few-shot learning are arguably MAML [14], a model-agnostic approach that uses a two-step gradient optimization process, and PROTONET [32], which learns an embedding space on which classes can be more easily differentiated. However, these methods and their improvements (such as Siamese Network, [25]) generally perform suboptimaly on the XML few-shot task due to its large number of independent labels. A few approaches have been proposed to tackle this problem. For instance, PfastreXML [17] addresses the rarity of the new labels by designing a loss that emphasizes tail labels. From a different perspective, DECAF [24] leverages the new label features to improve the learning of the tail distribution. However, while these models achieve reasonable performance on few-shot tasks, they generally yield suboptimal results in large XML problems endowed with taxonomies. On the other hand, our contribution, TAMLEC, is particularly well-suited for this setting, as its modular design allows for rapid adaptation by simply adding the corresponding task-specific layers, resulting in fast and efficient training, as illustrated by our experiments (see Section 4.4).

# 3 Method

This section introduces the main contributions of this paper, i.e. Taxonomy-Aware Tasks (TATs) and TAMLEC.

## 3.1 Weak-Semilattice and Taxonomy-Aware Tasks

Throughout this paper, we assume that the XML problem comes with a taxonomy $T$ that can be modeled as a Weak-Semilattice. We start by briefly recalling a few definitions.

Definition 1 (Partially Ordered Set). *Let $T$ be a set endowed with a binary relation $\leq$. Then $T$ is a partially ordered set (Poset), if $\leq$ is transitive, reflexive and antisymmetric.*

In the context of a taxonomy $T$, the binary relation generally represents a hierarchical relationship: the fact that a label $\ell_1 \in T$ is more general than a label $\ell_2 \in T$ will be denoted as $\ell_1 \leq \ell_2$. For example, if $T$ represents a scientific label taxonomy (see Figure 1 for a toy example), $\ell_1$ = NLP and $\ell_2$ = LLMs, then $\ell_1 \leq \ell_2$.

Definition 2 (Weak-Semilattice). *A partially ordered set $(T, \leq)$ is called a Weak-Semilattice if*

$$\forall T' \subset T,\ \exists \ell \in T \text{ such that } \forall \ell' \in T', \quad \ell \leq \ell'$$

*The set of elements that are smaller than $T'$, called the lower set of $T'$, is noted $\text{low}_T(T')$.*

As a consequence, any two elements of $T$ have at least one common lower bound in $T$. For a hierarchical taxonomy $T$, this lower bound can be seen as a label that is more general than any label in $T'$. In the toy taxonomy of scientific labels depicted in Figure 1, the greatest lower bound of Vocabulary and Machine Learning would be Computer Science.

***Relation with other structures.*** It is easy to see that trees are a special case of Weak-Semilattice, by defining $\ell_1 \leq \ell_2$ if and only if $\ell_1$ is an ancestor of $\ell_2$. Indeed, the root of the tree is always a lower bound and thus satisfies Definition 2. The inverse is not true, as illustrated by Figure 1, since some elements of the Weak-Semilattice (in this case the label "LLMs") can have multiple parents. We argue that this setting is quite natural in document labeling, as fine-grained labels can inherit from many general labels. For instance, multidisciplinary scientific topics can stem from multiple research fields, and the same phenomenon can be observed in many taxonomies [5, 31]. Importantly, Hierarchical Taxonomy-based methods such as Hector [26] rely on the tree structure to achieve their label prediction. This is not the case of our algorithm, TAMLEC, that is designed to accommodate any Weak-Semilattice structure. Similarly, note that Semilattices are a special case of Weak-Semilattice, as they require the existence of an unique infimum [4]. Finally, the relation between Weak-Semilattices and Posets is summarized as follows :

Lemma 1. *Let $(T, \leq)$ be a Poset. Then $(T, \leq)$ has a Condorcet winner if and only if $(T, \leq)$ is a Weak-Semilattice*

Proof. Let $(T, \leq)$ be a Poset. If $(T, \leq)$ has a Condorcet winner $c$, then $\forall T' \subset T, \forall \ell \in T', c \leq \ell$. Hence $(T, \leq)$ is a Weak-Semilattice. Conversely, if $(T, \leq)$ is a Weak-Semilattice, let $T' = T$, then by Definition 2 $\text{low}_T(T) \neq \emptyset$. Let $c \in \text{low}_T(T)$, then $c$ is a Condorcet winner. □

***Children and Width of a Weak-Semilattice.*** We also extend the notions of children and width to Weak-Semilattices, as they are important to TAMLEC training.

Definition 3 (Children in a Weak-Semilattice). *Let $(T, \leq)$ be a Weak-Semilattice and $\ell_1 \in T$. Let $\ell_2 \in T$ such that $\ell_1 \leq \ell_2$ and $\ell_1 \neq \ell_2$. $\ell_2$ is called a child of $\ell_1$ (denoted $\ell_1 \prec \ell_2$) if and only if*

$$\forall \ell \in T, \quad \text{if } \ell_1 \leq \ell \leq \ell_2 \text{ then } \ell_1 = \ell \text{ or } \ell_2 = \ell.$$

Definition 4 (Width of a Weak-Semilattice). *Let $(T, \leq)$ be a Weak-Semilattice. The width of the Weak-Semilattice $w_T$ is*

$$w_T = \max_{\ell \in T} \# \left\{ \ell' \in T, \quad \ell \prec \ell' \right\}$$

In other words, the width of $T$ is the maximal number of children of any element of $T$. This notion directly relates to the difficulty of predicting a path on $T$, as it represents the maximum number of labels to choose from when extending the path, and is key to the TAT-adapted loss used to train TAMLEC (see Section 3.2).

***Taxonomy-Aware Tasks.*** At the heart of our approach is the decomposition into Taxonomy-Aware Tasks (TATs). Such a decomposition allows to better leverage the Parallel Feature Sharing framework of the Multi-Task paradigm by adjusting the task-specific components of TAMLEC. Furthermore, as each resulting task contains significantly fewer labels, and thus a better document / label ratio, this approach alleviates the data scarcity problem. To achieve these advantages, we need to ensure that the splitting of the taxonomy must be coherent with the Weak-Semilattice structure, and not remove valuable information regarding the relation between labels, such as paths between multiple sub-tasks.

Definition 5 (Taxonomy-Aware Tasks). *Let $(T, \leq)$ be a Weak-Semilattice. Then the collection of sets $(T_1, \ldots, T_N)$ are Taxonomy-Aware Tasks if and only if:*

(1) $\forall 1 \leq i \leq N$, $T_i \subset T$ *and* $(T_i, \leq)$ *is a Weak-Semilattice,*
(2) $\forall 1 \leq i, j \leq N$, *if* $T_i \subset T_j$ *then* $T_i = T_j$,
(3) $\forall 1 \leq i \leq N$, $\forall \ell \in T_i$, $\forall \ell' \in T$, *if* $\ell \leq \ell'$ *then* $\ell' \in T_i$,
(4) $\forall \ell \in T \setminus low_T(T)$, $\exists 1 \leq i \leq N$ *such that* $\ell \in T_i$.

(1) and (3) enforce the preservation of paths: if one label $\ell$ is present in $T_i$ then all the paths that originate from $\ell$ are also in $T_i$. This is key to TAMLEC as it predicts labels by forecasting paths in the Weak-Semilattice structure. The second condition prevents repeated tasks, while the fourth ensures that all the labels, with the exception of the Condorcet winner, are part of at least one task.

Figure 2 shows an example of a Weak-Semilattice and a TATs decomposition. Note that the combination of the conditions in Definition 5 makes TATs decompositions unique. In the case where $T$ is a tree, Definition 5 implies that the TATs decomposition will be made of all the subtrees of $T$ whose roots are node of depth one (i.e. second level of the taxonomy). While in the tree setting the TATs are disjoints (i.e. they share no common labels), this is generally not true for Weak-Semilattices (see Figure 2).

## 3.2 TAMLEC

The main contributions behind TAMLEC are twofold. First, TAMLEC uses a modified transformer to predict paths in the taxonomy, which are then combined to obtain the final predictions. Unlike

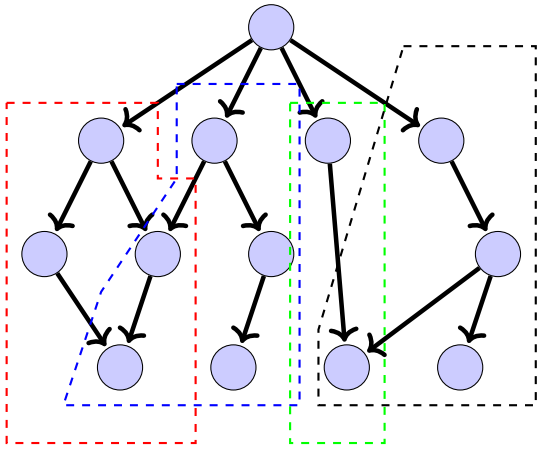


**Figure 2: Example of a Weak Semilattice taxonomy with a Taxonomy-Aware Tasks decomposition (polygons).**

previous approaches such as HECTOR [26], which assume that the label taxonomy is a tree, TAMLEC generalizes to weak-semilattice structures (see Section 3.1), which is a more flexible and more realistic framework that extends trees by, e.g., allowing multiple parent nodes per label. Second, TAMLEC leverages the TATs decomposition, introduced above, to partition the overall taxonomy into smaller, coherent substructures satisfying specific properties (see Definition 5). This enables TAMLEC to use a multi-task setup, and to leverage this decomposition by applying parallel feature sharing across tasks, while maintaining task-specific components (see Model Architecture below). Combined with a novel, task-adaptive training loss (see equation 1), this approach allows TAMLEC to substantially outperform state-of-the-art XMLC methods across several benchmark datasets (see Section 4).

***Path Prediction.*** While the labels of a document are generally encoded as a set, they can also be represented as a collection of paths in the taxonomy, from the most general to the more specific [26]. In the context of a Weak-Semilattice, a path is defined as a sequence of labels and their children, i.e. $\ell_1 \prec \ell_2 \prec \ldots \prec \ell_K$, where $\ell_1$ is the Condorcet winner, or "root", of the taxonomy (see Lemma 1). Such a path can be seen as a sequence of increasingly specific labels that characterize the documents.

Predicting paths has multiple advantages over predicting a set of labels. First, these paths yield a natural sequential structure, contrarily to sets, thus allowing to fully leverage the transformer architecture, which has been shown to achieve state-of-the-art performance on XML problems [26, 42]. Second, this approach naturally embeds the taxonomy structure into the prediction, as at each step, only the children of a label will be considered to be added to the path. This both encodes the relation between labels (a key ingredient of successful XML methods, see e.g. [43]), and also alleviates the label scale problem by strongly reducing the number of candidate at each step. A key difference with previous works is that TAMLEC predicts paths on TATs, which are sub-taxonomies of $T$. Depending on the path prefix, one or more TATs may be relevant to the predictions at hand. Thus TAMLEC predicts paths in parallel

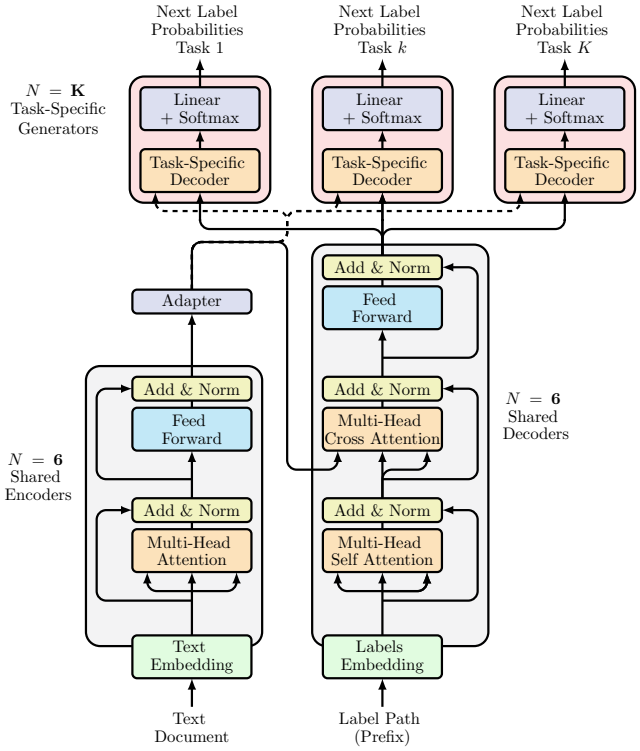



**Figure 3: TAMLEC's architecture. The model is made of 6 Encoders and 6 Decoders, with weights shared across tasks, as well as $K$ Task-Specific Generators (in light red), whose weights are task-specific (here $K = 3$). The task-specific generators are key to TAMLEC TATs decomposition approach.**

across all the relevant TATs. Moreover, even when only one TAT is compatible with a given label, multiple paths may lead to this label. The combination of predictions across multiple paths and TATs is discussed below.

***Model Architecture.*** Figure 3 details TAMLEC's architecture, which is based on Transformers [37]. Similar to HECTOR [26], TAMLEC uses both Decoder and Encoder blocks, whose weights are shared across all tasks. However, and in contrast to previous XML Transformer-based architectures, TAMLEC also introduces a new type of block, the Task-Specific Generator. When predicting the next label of a path, TAMLEC receives as input the text of the document, the current path, as well as the current TAT of interest. The document is first encoded using a 300 dimensional embedding, which is then fed into a stack of six Encoder blocks. Each Encoder block contains a multi-head attention layer, with 12 Heads, followed by a fully connected feed-forward layer with residual connection. At the end of the six Encoders blocks, the resulting 300 dimensional encoding is then projected into a 600 dimensional encoding using a fully connected adapter layer. In parallel, the label path is encoded using a 600 dimensional embedding, and then processed through a stack of six Decoder blocks. Each Decoder block is made of a multi-head self attention layer, with 12 Heads, followed by a cross attention layer, which captures the dependency between the document encoding and the path encoding, followed in turn by a fully connected feed-forward layer with a residual connection. Finally, at the end of the six Decoder blocks, the resulting 600 dimensional encoding is then sent to the Task-Specific Generator block corresponding to the TAT of interest. This block contains a task-specific decoder block, where labels in the label path that do not belong to the TAT are masked. This allows to focus on TAT-relevant label encoding. This layer is followed by a fully connected feed forward layer that maps to the space of all possible labels that belong to the TAT. Before training, the weights of TAMLEC's layers are initialized randomly, with the exception of the text embeddings where we use pretrained GloVe embeddings [28].

***Preprocessing data.*** In the following, we assume that the set of labels for each document $\mathcal{D}$ is complete, i.e. $\mathcal{D}$ contains the labels necessary to form a path to each label of $\mathcal{D}$. This is in line with the usual label completion assumption: if a document has a specific label $\ell$, it should also possess broader labels that contain the sub-label $\ell$ [26, 31]. If this assumption is not satisfied, the missing labels are added during the preprocessing of the dataset, in line with the Hierarchical Label Set Expansion proposed by [16]. In case a label $\ell \in \mathcal{D}$ has no valid path in $\mathcal{D}$ but multiple possible paths exist in the full taxonomy $T$, we add the minimum number of labels possible to obtain at least one valid path, with ties being broken at random. Then, each document's collection of labels $\mathcal{D}$ is transformed into a collection of paths $\mathcal{P}$. Each path $p \in \mathcal{P}$ is subsequently associated to one or more relevant tasks. A TAT $T_i$ is said to be relevant to a path $p$ if there exists a label $\ell \in p$ such that $\ell \in T_i$, i.e. if it contains at least one label of this path.

***Training Loss.*** To train TAMLEC we introduce a new TAT-dependent loss function $\mathcal{L}$. This loss is based on cross-entropy with label smoothing set to a value $\varepsilon = 0.01$ [34], and a confidence decreasing proportionally to the width of the relevant task $w_{T_i}$. Formally, for a given document $d$, a path prefix $p$, next label $\ell$, relevant task $T_i$ and predicted probability distribution over the next label $\hat{\ell}$, the loss is defined as:

$$\mathcal{L}(\hat{\ell}, \ell, T_i) = \left(1 - \varepsilon \frac{w_{T_i}}{1 + w_{T_i}}\right) \log\left(P(\hat{\ell} = \ell)\right) + \sum_{\ell' \in T_i, \ell' \neq \ell} \varepsilon \frac{w_{T_i}}{1 + w_{T_i}} \log\left(1 - P(\hat{\ell} = \ell')\right) \quad (1)$$

It is important to note that $\mathcal{L}$ only considers labels in the task $T_i$ and ignores other labels. Compared to the original label smoothing, the loss function is designed to be task-aware, by considering the difficulty and structure of each TAT. Indeed, it applies dynamic smoothing, where high confidence predictions are more favored for tasks with low width, compared to tasks with large width, as the loss is weighted for of each TAT using its width $w_{T_i}$, which is computed from the taxonomy (see Definition 4). Intuitively, this weighting decreases the label smoothing when the task has a small width – i.e. a limited number of labels to choose from at each level of the weak-semilattice, entailing more confidence in the model prediction. Conversely, when the width of the TAT is large, – i.e. a large number of possible labels – the model is encouraged to be more conservative.

| | MAG-CS | PubMed | EURLex |
|---|---|---|---|
| N Docs | 140994 | 331720 | 172120 |
| N Labels | 2641 | 5911 | 4492 |
| Avg Labels per Doc | 4.4 | 18.5 | 10.4 |
| Taxonomy Width | 145 | 21 | 145 |
| N TATS | 24 | 8 | 145 |
| Avg TAT Width | 10.5 | 15.5 | 3 |
| Med. Doc per TAT | 2443.33 | 34563 | 520 |

**Table 1: Important Datasets and TATs decomposition statistics. The three datasets exhibit very different TATs decomposition profiles, in term of number of TATs, width, and number of documents per TAT.**

***Training TAMLEC.*** The training of TAMLEC is performed in two steps. First, all weights are updated until the loss $\mathcal{L}$ stops improving on the validation set. Then, all shared layers (the 6 Decoders and 6 Encoders) are frozen, while the task-specific weights are fine-tuned on each task independently. Similarly, when new labels are introduced (XML few-shot setting), only the weights that are specific to this task are trained, resulting in fast and efficient convergence (as most of the weights are frozen), see Section 4.4.

***Label Prediction.*** At inference time, TAMLEC receives a document and an incomplete set of labels $\mathcal{D}$. Similarly to the training phase, this set of labels is transformed into a collection of paths $\mathcal{P}$ with their corresponding relevant TATs. For each path and TAT, TAMLEC generates multiple path extensions using beam search, a commonly-used algorithm for decoding structured predictors, similarly to [26]. Beam search maintains a list of the most promising candidate paths, together with their combined predicted probabilities, and iteratively updates it by feeding the paths into TAMLEC, until no further path can be obtained as the most specific labels have been reached. At the end, each label $\ell$ of the TAT is attributed a score equals to the sum of all the probabilities of the Beam search's predicted paths that had $\ell$ as a leaf. Scores are then summed across all paths and relevant TATs to obtain the final score of each label, which is then used to compute the final label ranking.

# 4 Experimental Evaluation

## 4.1 Experimental Setting

***Datasets.*** We perform our experiments on three commonly used XML datasets that are endowed with rich taxonomies: MAG-CS, PubMed, and EURLex. All the datasets are completed to abide by the hierarchical taxonomy using the process described in Section 3.2. The first dataset, MAG-CS, the Microsoft Academic Graph (MAG) Computer Science (CS) is a subset of the MAG dataset [38] focused on CS, which contains papers published between 1990 and 2020 at top CS conferences [43]. Example of the most general concepts include "Machine Learning", "Natural Language Processing" and "World Wide Web". The second dataset, PubMed, was published by [43] and contains scientific papers from top medical journals published between 2010 and 2020, endowed with labels from the Medical Subject Headings (MeSH) taxonomy. The most general concepts of the taxonomy include "Anatomy", "Psychiatry and Psychology" as well as "Diseases". Importantly, the MeSH taxonomy is particularly representative of a non-tree taxonomy, as many labels inherit from multiple parents. For instance, "Digestive System Neoplasms" can be reached through two distinct paths that include either "Digestive System Diseases" or "Neoplasms". Finally, the last dataset, EURLex [6], includes EU legislative documents equipped with labels from the European Vocabulary (EuroVoc) taxonomy. Table 1 summarizes the key datasets' characteristics, as well the description of their TATs decomposition. Interestingly, these three datasets' TATs decompositions are very different: for instance, EURLex TATs have an average width of 3, compared to more than 10 for the two other datasets. Furthermore, PubMed TATs decomposition contains only 8 TATs, compared to 145 for EURLex. This diversity allows testing TAMLEC on substantially different use-cases.

***Baseline Models.*** We compare the performance of TAMLEC with multiple recent XML baselines:

- MATCH [43] is a transformer-based approach that incorporates both document metadata and label hierarchy into the learning process. It enhances the objective function with a taxonomy-driven regularization, and enriches the document embeddings by using auxiliary metadata such as authors and publication venue. We retain the original architecture, consisting of a three-layer transformer encoder with two attention heads per layer and eight classification tokens.
- XML-CNN [21] is a deep learning-based algorithms for Extreme Multi-Label Classification. It uses a convolutional neural network (CNN) architecture [20] with dynamic max pooling—a strategy that preserves multiple salient features. We adopt the original configuration, consisting of a three-layer 1D CNN with convolutional filters of window sizes 2, 4, and 8, each retaining 128 features. The bottleneck layer is set to a dimensionality of 512, with a dropout rate of 0.5.
- AttentionXML [41] is a method that constructs a shallow and wide probabilistic label tree. It combines bi-directional LSTMs with attention mechanisms and fully connected layers to model label dependencies effectively. Our implementation is based on the official code released by the authors.
- FastXML [29] is a non-neural, tree-based method that partitions the document space using node-level optimization guided by the normalized discounted cumulative gain (nDCG) objective. In our experiments, we used the PfastreXML variant [18], configuring it with 50 trees, a maximum of 200 data points per leaf, up to 200 labels per leaf, and a hinge loss objective regularized with an L2 penalty.
- HECTOR [26], a recent state-of-the-art XMLCo algorithm which uses a modified transformer architecture to predict label paths in the taxonomy. It is important to note that the taxonomies considered in this experiment cannot be fully encoded by a tree – a requirement for HECTOR predictions. Thus, we modify the taxonomies provided to HECTOR by removing the minimum amount of relations possible to reduce each weak-semilattice to a tree, in line with the original experiments of HECTOR.

For the XML few-shot experiment (see Section 4.4), we compare TAMLEC to four few-shot methods.

| | Method | Precision | | | | NDCG | | | | F1 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | @1 | @2 | @3 | @4 | @2 | @3 | @4 | @5 | @1 | @2 | @3 | @4 |
| MAG-CS | MATCH | 0.79 | 0.77 | 0.75 | 0.74 | 0.79 | 0.78 | 0.79 | 0.78 | 0.65 | 0.65 | 0.66 | 0.68 |
| | XML-CNN | 0.52 | 0.56 | 0.56 | 0.57 | 0.58 | 0.61 | 0.63 | 0.64 | 0.41 | 0.46 | 0.49 | 0.52 |
| | Attention-XML | 0.81 | 0.79 | **0.78** | **0.78** | 0.81 | 0.81 | 0.81 | 0.81 | 0.67 | 0.67 | 0.69 | **0.71** |
| | Fast-XML | 0.49 | 0.54 | 0.55 | 0.57 | 0.56 | 0.60 | 0.64 | 0.66 | 0.37 | 0.44 | 0.48 | 0.52 |
| | HECTOR | 0.83 | 0.76 | 0.74 | 0.72 | 0.79 | 0.77 | 0.77 | 0.76 | 0.69 | 0.65 | 0.65 | 0.66 |
| | TAMLEC | **0.87** | **0.80** | **0.78** | 0.77 | **0.83** | **0.81** | **0.81** | **0.81** | **0.73** | **0.69** | **0.69** | 0.70 |
| PUBMed | MATCH | 0.79 | 0.75 | 0.76 | 0.78 | 0.76 | 0.77 | 0.79 | 0.81 | 0.28 | 0.42 | 0.49 | 0.52 |
| | XML-CNN | 0.75 | 0.70 | 0.70 | 0.73 | 0.71 | 0.72 | 0.75 | 0.78 | 0.26 | 0.38 | 0.44 | 0.48 |
| | Attention-XML | 0.79 | 0.76 | 0.76 | 0.78 | 0.76 | 0.77 | 0.80 | 0.82 | 0.28 | 0.43 | 0.49 | 0.52 |
| | Fast-XML | 0.73 | 0.69 | 0.70 | 0.75 | 0.70 | 0.72 | 0.77 | 0.81 | 0.24 | 0.37 | 0.43 | 0.48 |
| | HECTOR | 0.84 | 0.80 | 0.79 | 0.80 | 0.81 | 0.81 | 0.82 | 0.83 | 0.31 | 0.47 | 0.52 | 0.54 |
| | TAMLEC | **0.87** | **0.84** | **0.84** | **0.86** | **0.84** | **0.85** | **0.87** | **0.89** | **0.32** | **0.49** | **0.55** | **0.58** |
| EURLex | MATCH | 0.85 | 0.89 | 0.85 | 0.84 | 0.89 | 0.86 | 0.85 | 0.79 | 0.75 | 0.84 | 0.81 | 0.81 |
| | XML-CNN | 0.84 | 0.88 | 0.85 | 0.84 | 0.89 | 0.86 | 0.85 | 0.80 | 0.74 | 0.84 | 0.81 | 0.81 |
| | Attention-XML | 0.87 | 0.89 | 0.86 | 0.84 | 0.90 | 0.87 | 0.86 | 0.81 | 0.76 | 0.85 | 0.82 | 0.82 |
| | Fast-XML | 0.83 | 0.89 | 0.89 | 0.89 | 0.90 | 0.91 | 0.89 | 0.82 | 0.72 | 0.84 | 0.85 | 0.82 |
| | HECTOR | 0.89 | 0.88 | 0.85 | 0.80 | 0.89 | 0.86 | 0.82 | 0.75 | 0.78 | 0.84 | 0.81 | 0.78 |
| | TAMLEC | **0.94** | **0.95** | **0.93** | **0.90** | **0.96** | **0.94** | **0.91** | **0.85** | **0.83** | **0.91** | **0.89** | **0.87** |

**Table 2: Performance comparison of TAMLEC and other competing methods for XMLCo on the MAG-CS, PubMed and EURLex datasets. The best values for each combination of metric and dataset are written in bold.**

- MAML-T, based on the few-shot Meta Learning method MAML [14], which consists of two main steps: (i) a learning part, where a *student* model is trained on a series of tasks, in our case represented by the TATs, and (ii) a meta-learning part, where a *teacher* model optimizes the *student* model depending on its performance on the different tasks.
- Protonet-T [32] is a few-shot metric-based method that performs classification from a clustering of the data points. By seeing a few examples of each class, the model computes a centroid, i.e. a prototype in the embedding space for each class, and classifies new data points based on the Euclidean distance to the prototypes. In this experiment, the document embeddings result from a three layer encoder-only neural network with two multi-heads self attention layers (see below), and the centroids are the average of the document embeddings of the same class.
- Deep Brownian Distance Covariance (BDC) [39] is a recent few-shot method with a similar approach to Protonet. Instead of getting the document embeddings from a transformer model as in Protonet-T, the embeddings are computed from the joint probability density function based on the raw document embeddings [35, 36]. This computation can be performed in an isolated pooling layer in a neural network as shown by the authors.
- Siamese Networks and Label Tuning (SIAM) [25], a recent few-shot algorithm, tunes precomputed label embeddings instead of document embeddings. This approach is quicker and more efficient as the weights of the pretrained model are not updated during the tuning stage. We pretrained a model with the MATCH algorithm, and then used SIAM in the fine-tuning part for the few-shot experiment.

Since these few-shot algorithms require a base model adapted to the problem, we use a modified transformer architecture derived from MATCH, i.e. a three layer encoder-only neural network with two multi-head self attention layers. All baselines were trained on our modified versions of the datasets, using the hyperparameter values recommended by their respective authors. Finally, TAMLEC was trained using the Adam [19] optimizer with an initial learning rate of $5 \times 10^{-5}$, a weight decay of $10^{-2}$ and a smoothing parameter of 0.01. In addition to the different baselines, we also perform an ablation study of the XMLCo performance of TAMLEC (see Section 4.3). All experiments were run on a machine equipped with a Tesla V100 GPU, using python 3.11 and pytorch [27]. An implementation of TAMLEC can be found on github[3].

***Evaluation and Metrics.*** Throughout our experiments, we evaluate the different methods using their ranked label prediction $\mathcal{R}$, which lists the labels by decreasing order of predicted probability. In other words, $\forall n > 0\ \mathcal{R}(n)$ is the $n$-th most likely label according to the predictions. We compare the various results using three commonly used metrics [26]: Precision at $k$ ($P@k$), Normalized Discounted Cumulative Gain at $k$ ($NDCG@k$) and F1 at $k$. Formally these metrics are defined as follows. Let $y_n = 1$ if $\mathcal{R}(n)$ is correct and 0 otherwise. In other words, $y_n$ is a boolean variable that indicates whether the $n$-th element of $\mathcal{R}$ belongs to the document. With these notations, $P@k$ is defined as:

$$P@k = \frac{1}{k} \sum_{n=1}^{k} y_n,$$

[3] https://github.com/Jythen/Tamlec

| | Method | Precision ↑ | | | | NDCG ↑ | | | | F1 ↑ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | @1 | @2 | @3 | @4 | @2 | @3 | @4 | @5 | @1 | @2 | @3 | @4 |
| MAG-CS | $\sqrt{\text{TAMLEC}}$ | 0.83 | 0.77 | 0.73 | 0.71 | 0.79 | 0.77 | 0.76 | 0.75 | 0.69 | 0.66 | 0.65 | 0.65 |
| | TAMLEC | **0.87** | **0.80** | **0.78** | **0.77** | **0.83** | **0.81** | **0.81** | **0.81** | **0.73** | **0.69** | **0.69** | **0.70** |
| | TAMLEC-D | 0.84 | 0.77 | 0.74 | 0.71 | 0.79 | 0.77 | 0.76 | 0.75 | 0.70 | 0.66 | 0.65 | 0.65 |
| | TAMLEC-W | 0.84 | 0.77 | 0.73 | 0.72 | 0.79 | 0.77 | 0.76 | 0.75 | 0.70 | 0.66 | 0.65 | 0.66 |
| PUBMed | $\sqrt{\text{TAMLEC}}$ | 0.84 | 0.80 | 0.80 | 0.81 | 0.81 | 0.81 | 0.82 | 0.83 | 0.31 | 0.47 | 0.53 | 0.55 |
| | TAMLEC | **0.87** | **0.84** | **0.84** | **0.86** | **0.84** | **0.85** | **0.87** | **0.89** | **0.32** | **0.49** | **0.55** | **0.58** |
| | TAMLEC-D | 0.84 | 0.80 | 0.79 | 0.80 | 0.81 | 0.81 | 0.82 | 0.83 | 0.31 | 0.47 | 0.52 | 0.55 |
| | TAMLEC-W | 0.84 | 0.80 | 0.79 | 0.80 | 0.81 | 0.81 | 0.82 | 0.83 | 0.31 | 0.47 | 0.52 | 0.55 |
| EURLex | $\sqrt{\text{TAMLEC}}$ | 0.91 | 0.91 | 0.86 | 0.81 | 0.91 | 0.88 | 0.82 | 0.77 | 0.80 | 0.86 | 0.83 | 0.78 |
| | TAMLEC | **0.94** | **0.95** | **0.93** | **0.90** | **0.96** | **0.94** | **0.91** | **0.85** | **0.83** | **0.91** | **0.89** | **0.87** |
| | TAMLEC-D | 0.92 | 0.91 | 0.87 | 0.82 | 0.92 | 0.88 | 0.83 | 0.77 | 0.81 | 0.87 | 0.83 | 0.79 |
| | TAMLEC-W | 0.91 | 0.91 | 0.86 | 0.81 | 0.91 | 0.87 | 0.84 | 0.78 | 0.80 | 0.86 | 0.83 | 0.79 |

**Table 3: Ablation study of TAMLEC on the XMLCo experiment on the MAG-CS, PubMed and EURLex datasets. The best values for each combination of metric and dataset are written in bold.**

. i.e. the average number of correctly predicted labels among the first $k$ elements of $\mathcal{R}$. $NDCG@k$ provides a smoother measurement of the quality of the ranking $\mathcal{R}$, by assigning lower weights to failed predictions in the tail of the ranking. Formally,

$$NDCG@k = \frac{\sum_{n=1}^{k} \frac{y_n}{\log(n+1)}}{\sum_{n=1}^{\min(k,k_y)} \frac{1}{\log(n+1)}},$$

where $k_y$ is the number of labels of the document. Finally, F1@k is defined as the harmonic mean of the precision@k and the recall@k. All metrics ($@k$) are reported as average on the test set across all the documents that are equipped with at least $k$ labels.

## 4.2 Label Completion

***Experimental design.*** In the first set of experiments, we simulate an XMLCo problem by removing all the labels from each document except the ones associated with the most general concepts in the taxonomy. The motivation behind this choice is twofold: first, the most general labels are the easiest to obtain[4] and are the most commonly present in a dataset [26, 31]. Second, due to our hierarchical assumption, more general labels can easily be deduced from more specific ones. To use the different baselines for XMLCo, we run a normal inference step and then skip model predictions of labels that already belong to the document.

***Results.*** Table 2 reports the results of the XMLCo experiments. First, we note that TAMLEC performs significantly better than the other XML methods (MATCH, XML-CNN, FastXML, AttentionXML and HECTOR) across almost all metrics and datasets, and at worst performs as good as other baselines (such as for $k = 3$ on MAG-CS). The advantage of TAMLEC is particularly visible on EURLex, where it outperforms the other baselines by a wide margin. This can be explained by the fact that the EURLex TATs decomposition contains more than 100 TATs, significantly more than the other datasets, maximizing the benefits of TAMLEC TATs approach. In contrast, the advantage of TAMLEC is the smallest for PUBMed, which only contains 8 TATS. These results highlight the impact of the TATs decomposition on TAMLEC. Finally, it is interesting to note that FastXML significantly underperforms other methods, in particular on MAG-CS. As the only non neural network-based method of our benchmark, these results point to the clear advantage of deep learning methods for XMLC.

[4]for instance, the venue at which a scientific paper is published often suffices to deduce its field of research.

## 4.3 Ablation analysis

***Experimental design.*** To evaluate the importance and effectiveness of each of the main components of TAMLEC, we perform an in-depth ablation study. In particular, we use three truncated versions of our algorithm:

- $\sqrt{\text{TAMLEC}}$, a version of TAMLEC that only includes the TATs decomposition, but without adaptive loss (i.e. we set $w_{T_i} = 1$ for all tasks) nor task-specific decoder layer (i.e. the last shared decoder is directly followed by a task-specific linear+softmax layer),
- TAMLEC-D, which includes the TATs decomposition and the task-specific decoder, without the width adaptive loss,
- TAMLEC-W, which includes the TATs decomposition and the width adaptive loss, but no task-specific decoder.

***Results.*** The results are presented in Table 3. We observe that all the truncated versions are significantly worse than TAMLEC across all metrics and datasets. Interestingly, the performance of $\sqrt{\text{TAMLEC}}$, TAMLEC-D and TAMLEC-W are very similar, which indicates that individual components may not bring significant advantage. By contrast, TAMLEC outperforms its tampered down version significantly, hinting at the importance of all the key components and their synergies (TATs decomposition, adaptive loss, and task-specific layers).

## 4.4 Few-Shot XML

***Experimental design.*** In this experiment, we aim at evaluating the XML few-shot potential of TAMLEC compared to different

| | Algorithm | Global Precision ↑ | | Global NDCG ↑ | | Global F1 ↑ | | NT Precision ↑ | | NT NDCG ↑ | | NT F1 ↑ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | @1 | @3 | @2 | @4 | @1 | @3 | @1 | @3 | @2 | @4 | @1 | @3 |
| MAG-CS | MATCH | 0.727 | 0.665 | 0.735 | 0.706 | 0.601 | 0.590 | 0.824 | **0.714** | **0.704** | 0.804 | 0.791 | 0.682 |
| | MAML-T | 0.305 | 0.283 | 0.347 | 0.367 | 0.215 | 0.244 | 0.715 | 0.636 | 0.695 | 0.637 | 0.672 | 0.606 |
| | SIAM | 0.788 | 0.760 | 0.770 | 0.788 | 0.646 | 0.677 | 0.747 | **0.714** | 0.609 | 0.820 | 0.702 | 0.671 |
| | PROTONET-T | 0.079 | 0.074 | 0.071 | 0.087 | 0.069 | 0.065 | 0.092 | 0.364 | 0.199 | 0.400 | 0.083 | 0.350 |
| | BDC | 0.055 | 0.050 | 0.051 | 0.057 | 0.048 | 0.043 | 0.057 | 0.278 | 0.213 | 0.287 | 0.048 | 0.261 |
| | $\sqrt{\text{TAMLEC}}$ | 0.832 | 0.730 | 0.768 | 0.779 | 0.689 | 0.666 | 0.941 | 0.704 | 0.665 | 0.820 | 0.891 | 0.684 |
| | TAMLEC | **0.865** | **0.773** | **0.825** | **0.812** | **0.717** | **0.686** | **0.959** | 0.712 | 0.703 | **0.822** | **0.910** | **0.675** |
| PUBMed | MATCH | 0.712 | 0.688 | 0.675 | 0.766 | 0.239 | 0.427 | 0.967 | 0.766 | 0.845 | 0.790 | 0.585 | 0.705 |
| | MAML-T | 0.544 | 0.504 | 0.526 | 0.608 | 0.154 | 0.272 | **0.990** | 0.602 | 0.769 | 0.607 | **0.606** | 0.553 |
| | SIAM | 0.786 | 0.764 | 0.769 | 0.812 | 0.280 | 0.498 | 0.965 | 0.681 | 0.794 | 0.662 | 0.595 | 0.636 |
| | PROTONET-T | 0.025 | 0.029 | 0.026 | 0.032 | 0.009 | 0.020 | 0.172 | 0.249 | 0.220 | 0.256 | 0.086 | 0.231 |
| | BDC | 0.026 | 0.030 | 0.027 | 0.030 | 0.010 | 0.021 | 0.183 | 0.206 | 0.208 | 0.213 | 0.106 | 0.185 |
| | $\sqrt{\text{TAMLEC}}$ | 0.820 | 0.803 | 0.804 | 0.782 | 0.302 | 0.508 | 0.962 | 0.827 | 0.921 | 0.829 | 0.583 | 0.730 |
| | TAMLEC | **0.863** | **0.833** | **0.840** | **0.867** | **0.320** | **0.549** | **0.990** | **0.876** | **0.925** | **0.907** | **0.606** | **0.803** |
| EURLex | MATCH | 0.770 | 0.785 | 0.844 | 0.789 | 0.673 | 0.785 | 0.920 | 0.858 | 0.902 | 0.791 | 0.598 | 0.878 |
| | MAML-T | 0.650 | 0.741 | 0.755 | 0.713 | 0.554 | 0.709 | 0.914 | 0.858 | 0.885 | 0.857 | 0.588 | 0.837 |
| | SIAM | 0.852 | 0.840 | 0.891 | 0.836 | 0.744 | 0.814 | 0.731 | 0.767 | 0.754 | 0.754 | 0.465 | 0.760 |
| | PROTONET-T | 0.514 | 0.594 | 0.614 | 0.632 | 0.446 | 0.568 | 0.437 | 0.643 | 0.449 | 0.710 | 0.264 | 0.627 |
| | BDC | 0.539 | 0.577 | 0.620 | 0.646 | 0.472 | 0.564 | 0.515 | 0.656 | 0.592 | 0.747 | 0.318 | 0.660 |
| | $\sqrt{\text{TAMLEC}}$ | 0.904 | 0.881 | 0.912 | 0.847 | 0.802 | 0.822 | 0.945 | 0.902 | 0.938 | 0.774 | 0.614 | 0.862 |
| | TAMLEC | **0.944** | **0.913** | **0.958** | **0.895** | **0.832** | **0.873** | **0.968** | **0.963** | **0.967** | **0.869** | **0.624** | **0.943** |

**Table 4: Results of the XML few-shot experiment. NT (resp. Global) indicates a metric computed on the new task (resp. all the tasks, including the new task). The best values for each combination of metric and dataset are written in bold.**

few-shot methods as well as MATCH without modification. We proceed as follows: during training, a TAT $T_i$ is withheld from the training set, by removing all the labels of $T_i$ as well as all the documents that are solely equipped with labels of $T_i$. After being trained on this large training set until convergence, each algorithm is presented with the new task for fine-tuning: this is performed either by training the models for a few epochs on the new task (MATCH), or by using the algorithm-specific few-shot training process (MAML-T, Protonet-T, SIAM, BDC). For TAMLEC, only the task-specific parameters associated to this new task are trained, resulting in extremely fast fine-tuning.

***Results.*** Table 4 reports the metrics on both the new task (NT) and all the tasks (Global). First, TAMLEC performance is much higher than other methods on the Global metrics. In fact, its performance are barely lower than in the regular experiment (Table 2), where all the data is present during preliminary training. This indicates that TAMLEC is able to adapt seamlessly to the emergence of a completely new task after the initial training, as most of the adaptation is done using the task-specific decoder block. Comparatively, the other XMLC method (MATCH) exhibits significantly worse global performance as it tries to adapt to the new task. Similarly, MAML-T and PROTONET-T achieve poor performance on the different XMLCo Global metrics. These results can be explained by the fact that XML is a notoriously difficult problem, for which MAML and PROTONET (the base meta-algorithms) were not designed. This is particularly true for PROTONET and BDC, whose cluster embedding approach is particularly challenged by the number of possible labels. This is not the case for SIAM (the combination of Siamese network and MATCH), which achieves reasonable performance on Global metrics. As the different methods were fined-tuned specifically on the new task, most algorithms were able to achieve very strong NT metrics (with the exception of PROTONET). While TAMLEC does not always achieve the best results, its metrics are always at least very close to the best performing method on the new task. Furthermore, and as discussed above, the better performance of other methods, such as MATCH on the new task, comes at a drastic cost to their global results. Thus, we argue that TAMLEC achieves the best trade-off between adapting to the new task and retaining performance on the entire problem.

## 5 Conclusion and Future Work

In this paper, we introduced a new algorithm, TAMLEC, that tackles XMLCo problems by dividing the associated taxonomy into Taxonomy-Aware Tasks (TATs), where each task is adapted to the path structure of the taxonomy. By leveraging ideas from parallel feature sharing, TAMLEC uses the TATs decomposition to achieve substantially better performance on label completion and XML few-shot problems across multiple datasets. This advantage scales with the complexity of the taxonomy: the more subtasks a taxonomy contains, the larger the advantage of TAMLEC. Future work includes the study of new types of TATs decomposition, where some TATs can be more specific (i.e. starting with more specific concepts) than others, and their integration into the method's architecture.

## 6 GenAI Usage Disclosure

GenAI was only used for small editing of the paper, i.e. checking for spelling and grammatical mistakes.

## References


[1] Isabelle Augenstein and Anders Søgaard. 2017. Multi-task learning of keyphrase boundary classification. *arXiv preprint arXiv:1704.00514* (2017).

[2] Danielle Caled, Mário J. Silva, Bruno Martins, and Miguel Won. 2022. Multi-label classification of legislative contents with hierarchical label attention networks. *Int. J. Digit. Libr.* 23, 1 (2022), 77–90. doi:10.1007/s00799-021-00307-w

[3] Rich Caruana. 1997. Multitask learning. *Machine learning* 28 (1997), 41–75.

[4] Ivan Chajda, Radomír Halaš, and Jan Kühr. 2007. *Semilattice structures*. Vol. 30. Heldermann Lemgo.

[5] Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. 2021. MultiEURLEX – A multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (Punta Cana, Dominican Republic). Association for Computational Linguistics. https://arxiv.org/abs/2109.00904

[6] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2019. Large-Scale Multi-Label Text Classification on EU Legislation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, Anna Korhonen, David R. Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, 6314–6322. doi:10.18653/v1/p19-1636

[7] Shuaichen Chang, Pengfei Liu, Yun Tang, Jing Huang, Xiaodong He, and Bowen Zhou. 2020. Zero-shot text-to-sql learning with auxiliary task. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 7488–7495.

[8] Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S. Dhillon. 2020. Taming Pretrained Transformers for Extreme Multi-label Text Classification. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*. ACM, 3163–3171.

[9] Olivier Chapelle, Pannagadatta Shivaswamy, Srinivas Vadrevu, Kilian Weinberger, Ya Zhang, and Belle Tseng. 2010. Multi-task learning for boosting with application to web search ranking. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1189–1198.

[10] Shijie Chen, Yu Zhang, and Qiang Yang. 2024. Multi-task learning in natural language processing: An overview. *Comput. Surveys* 56, 12 (2024), 1–32.

[11] Giuseppe Cuccu, Christophe Broillet, Carolin Reischauer, Harriet Thoeny, and Philippe Cudré-Mauroux. 2022. Typhon: Parallel Transfer on Heterogeneous Datasets for Cancer Detection in Computer-Aided Diagnosis. In *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 5223–5232.

[12] Giuseppe Cuccu, Johan Jobin, Julien Clément, Akansha Bhardwaj, Carolin Reischauer, Harriet Thöny, and Philippe Cudré-Mauroux. 2020. Hydra: Cancer detection leveraging multiple heads and heterogeneous datasets. In *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 4842–4849.

[13] Yuxiao Dong, Hao Ma, Zhihong Shen, and Kuansan Wang. 2017. A century of science: Globalization of scientific collaborations, citations, and innovations. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 1437–1446.

[14] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*. PMLR, 1126–1135.

[15] Francesco Gargiulo, Stefano Silvestri, and Mario Ciampi. 2019. Exploit hierarchical label knowledge for deep learning. In *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE, 539–542.

[16] Francesco Gargiulo, Stefano Silvestri, Mario Ciampi, and Giuseppe De Pietro. 2019. Deep neural network for hierarchical extreme multi-label text classification. *Appl. Soft Comput.* 79 (2019), 125–138. doi:10.1016/j.asoc.2019.03.041

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

[18] Himanshu Jain, Yashoteja Prabhu, and Manik Varma. 2016. Extreme Multi-label Loss Functions for Recommendation, Tagging, Ranking & Other Missing Label Applications. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. ACM, 935–944.

[19] Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[20] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. 1989. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems* 2 (1989).

[21] Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep Learning for Extreme Multi-label Text Classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*. ACM, 115–124.

[22] Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for Extreme Multi-label Text Classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku Tokyo Japan, 2017-08-07). ACM, 115–124. doi:10.1145/3077136.3080834

[23] Weiwei Liu, Xiaobo Shen, Haobo Wang, and Ivor W. Tsang. 2020. The Emerging Trends of Multi-Label Learning. *CoRR* abs/2011.11197 (2020).

[24] Anshul Mittal, Kunal Dahiya, Sheshansh Agrawal, Deepak Saini, Sumeet Agarwal, Purushottam Kar, and Manik Varma. 2021. Decaf: Deep extreme classification with label features. In *Proceedings of the 14th ACM international conference on web search and data mining*. 49–57.

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

[26] Natalia Ostapuk, Julien Audiffren, Ljiljana Dolamic, Alain Mermoud, and Philippe Cudré-Mauroux. 2024. Follow the Path: Hierarchy-Aware Extreme Multi-Label Completion for Semantic Text Tagging. In *Proceedings of the ACM on Web Conference 2024*. 2094–2105.

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703 [cs.LG] https://arxiv.org/abs/1912.01703

[28] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543. http://www.aclweb.org/anthology/D14-1162

[29] Yashoteja Prabhu and Manik Varma. 2014. FastXML: A Fast, Accurate and Stable Tree-Classifier for Extreme Multi-Label Learning. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2014-08-24) *(KDD '14)*. Association for Computing Machinery, 263–272. doi:10.1145/2623330.2623651

[30] 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*. ACM, 263–272.

[31] Miguel Romero, Felipe Kenji Nakano, Jorge Finke, Camilo Rocha, and Celine Vens. 2023. Leveraging class hierarchy for detecting missing annotations on hierarchical multi-label classification. *Comput. Biol. Medicine* 152 (2023), 106423. doi:10.1016/j.compbiomed.2022.106423

[32] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems* 30 (2017).

[33] Yisheng Song, Ting Wang, Puyu Cai, Subrota K Mondal, and Jyoti Prakash Sahoo. 2023. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *Comput. Surveys* 55, 13s (2023), 1–40.

[34] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2818–2826. doi:10.1109/CVPR.2016.308

[35] Gábor J Székely and Maria L Rizzo. 2009. Brownian distance covariance. (2009).

[36] Gábor J Székely, Maria L Rizzo, and Nail K Bakirov. 2007. Measuring and testing dependence by correlation of distances. (2007).

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 5998–6008.

[38] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. 2020. Microsoft Academic Graph: When experts are not enough. *Quant. Sci. Stud.* 1, 1 (2020), 396–413.

[39] Jiangtao Xie, Fei Long, Jiaming Lv, Qilong Wang, and Peihua Li. 2022. Joint distribution matters: Deep brownian distance covariance for few-shot classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 7972–7981.

[40] Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. 2023. To Repeat or Not To Repeat: Insights from Scaling LLM under Token-Crisis. arXiv:2305.13230 [cs] doi:10.48550/arXiv.2305.13230

[41] 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*. 5812–5822.

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

[43] Yu Zhang, Zhihong Shen, Yuxiao Dong, Kuansan Wang, and Jiawei Han. 2021. MATCH: Metadata-Aware Text Classification in A Large Hierarchy. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. ACM / IW3C2, 3246–3257.

[44] Yu Zhang and Qiang Yang. 2018. An overview of multi-task learning. *National Science Review* 5, 1 (2018), 30–43.

[45] Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE transactions on knowledge and data engineering* 34, 12 (2021), 5586–5609.