TagRec++

Hierarchical Label Aware Attention Network for Question Categorization

Viswanathan, Venktesh; Mohania, Mukesh; Goyal, Vikram

DOI
10.1109/TKDE.2024.3354504

Publication date
2024

Document Version
Final published version

Published in
IEEE Transactions on Knowledge and Data Engineering

TagRec++: Hierarchical Label Aware Attention Network for Question Categorization

Venktesh V, Mukesh Mohania, and Vikram Goyal

***Abstract*—Online learning systems have multiple data repositories in the form of transcripts, books and questions. To enable ease of access, such systems organize the content according to a well defined taxonomy of hierarchical nature (subject - chapter -topic). The task of categorizing inputs to the hierarchical labels is usually cast as a flat multi-class classification problem. Such approaches ignore the semantic relatedness between the terms in the input and the tokens in the hierarchical labels. Alternate approaches also suffer from class imbalance when they only consider leaf level nodes as labels. To tackle the issues, we formulate the task as a dense retrieval problem to retrieve the appropriate hierarchical labels for each content. In this paper, we deal with categorizing questions and learning content. We model the hierarchical labels as a composition of their tokens and use an efficient cross-attention mechanism to fuse the information with the term representations of the content. We also adopt an adaptive in-batch hard negative sampling approach which samples better negatives as the training progresses. We demonstrate that the proposed approach *TagRec++* outperforms existing state-of-the-art approaches on question and learning content datasets as measured by Recall@k. In addition, we demonstrate zero-shot capabilities of *TagRec++* and preliminary analysis of it's ability to adapt to label changes.**

***Index Terms*—Attention, contrastive learning, dynamic triplet sampling, hard-negatives, transformer.**

I. Introduction

ONLINE educational systems organize learning content like questions according to a hierarchical learning taxonomy of form subject-chapter-topic. For instance, a content about "pH level" is tagged with the learning taxonomy *"science - chemistry - acids"*. In the above example, *science* is the root node and *acids* is the leaf node. Organization of content in such standard format aids in better accessibility as users can easily navigate through large repositories of learning content by searching using different *facets* like the subject, chapter or topic names. The automated taxonomy tagger would aid in on-boarding content at scale from other sources by tagging them with a standardized learning taxonomy. Tagging content to a standardized taxonomy can be used for redirection of questions to relevant subject-matter experts in question answer forums of such systems. However, manual labeling of content with the hierarchical taxonomy is cumbersome. Automated tagging of content with learning taxonomy would enable indexing of content at scale and conserve time.

The Hierarchical Label Structure and Class Imbalance Problem: Automated approaches for the hierarchical categorization task must capture the relationship between content and the labels for effective content categorization. They must also preserve the hierarchical structure of the labels. However, existing approaches for tasks involving categorization of content to labels of hierarchical form usually cast it as *flat multi-class classification* problem [1], [2]. The flat classification approaches ignore the *hierarchical structure* in the label space and encode the labels as numbers. Alternate approaches [3], [4], [5] consider only the leaf nodes as labels to reduce the label space. In the former method the hierarchy is ignored and in the latter the problem of *class imbalance* occurs as most of the content is attached to a few leaf nodes. It has been demonstrated that contrastive learning helps address the class imbalance issue in other tasks and datasets [6], [7]. *TagRec++* follows a similar design philosophy to tackle the class imbalance problem.

Representation Learning and Extensibility: A major challenge in existing flat-classification approaches is that they are not easily extensible. This is because the learned representations, do not capture the hierarchical structure of the labels [8] and the relationship between the content and the hierarchical labels. Hence, the representations cannot be used for other downstream tasks like tagging other related content not seen during training or content retrieval based on taxonomy. Another challenge is the *open-set identification* problem, where new labels may emerge in the label space owing to addition of new topics or removal of old topics. The new hierarchical labels would still be semantically related to the old labels, and hence the model must be able to adapt by design to changes in the label space without re-training. Further, traditional multi-label multi-class classification approaches require changes in the model architecture and re-training to adapt to changes in the label space.

An Approach for Capturing Label Structure and Tackling Class Imbalance: To tackle the challenges mentioned previously, we propose an approach, *TagRec++*, that can capture the structure of the labels and relationship between the content and the labels. The problem is viewed as a dense retrieval task [9] where the goal is to retrieve the most relevant labels for a given question as shown in Fig. 1. Since the tokens in the label are abstractions of their word descriptions, they are related to the

Manuscript received 8 August 2022; revised 11 December 2023; accepted 31 December 2023. Date of publication 16 January 2024; date of current version 10 June 2024. This work was supported by Extramarks Education India Pvt. Ltd., SERB, FICCI (PM fellowship), Infosys Centre for AI and TiH Anubhuti (IIITD). Recommended for acceptance by Y. Gao. *(Corresponding author: Venktesh V.)*

Venktesh V is with the Tu Delft, 2628 CD Delft, The Netherlands (e-mail: venkteshv@iiitd.ac.in).

Mukesh Mohania and Vikram Goyal are with the Department of CSE, IIIT-Delhi, New Delhi, Delhi 110020, India (e-mail: mukesh@iiitd.ac.in; vikram@iiitd.ac.in).

Digital Object Identifier 10.1109/TKDE.2024.3354504

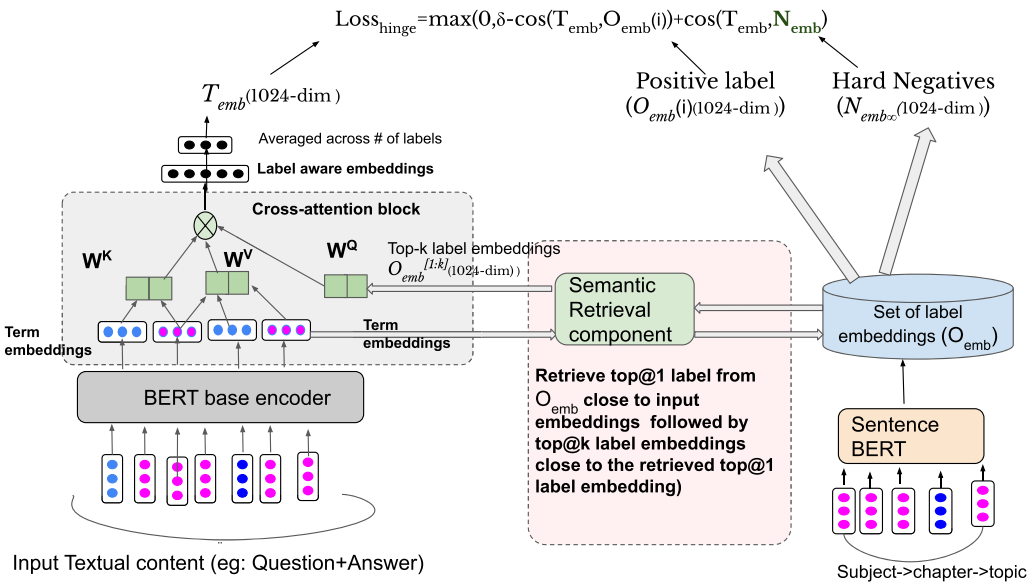

Fig. 1. Training loop of Proposed approach: TagRec++. The model is trained in a contrastive learning approach to align the input with the corresponding labels. O_{emb} refers to the embedding store for all hierarchical labels. N_{emb} refers to the hard negatives dynamically sampled during training for each training sample. δ is the margin value used in loss function.

terms in the content. Hence, we adopt a contrastive learning approach by projecting the content and labels to a continuous vector space to ensure the label representations are aligned with their corresponding content representations. We extend upon the work of TagRec [10] which proposed a simple two-tower architecture for bringing together the appropriate vector sub-spaces of the input and hierarchical labels closer.

The *TagRec++* fuses the content representations with the related hierarchical labels using an efficient interactive attention mechanism. It helps to better capture the relationship between the terms in the input learning content and the tokens in the hierarchical taxonomy. Specifically, it uses a late interaction approach where the label embeddings can be pre-computed and indexed, unlike cross-encoder based approaches.

The *TagRec++* uses a contrastive learning approach where triplets (anchor, positive, negative) are mined to pull apart the negatives from the anchor and bring the anchor and the positive sample closer. The sampling of negatives helps in learning better representations that have higher capacity to distinguish between positive and negative labels, thereby increasing recall during retrieval. Hence, *TagRec++* uses an adaptive in-batch hard-negative sampling approach where hierarchical labels closer to the content representation are chosen as hard-negatives. They are sampled dynamically in the training loop, further helping in disentangling the vector space of positive and negative labels.

Evaluation: The *TagRec++* is evaluated on datasets of question banks, long form video transcripts in science and related domains. The content may have both the question and its answer to capture more semantic context. The combined "question-answer" or the "question" in isolation is our learning content, and we will use the terms "content" and "question" interchangeably in the rest of the discussion.

Key Takeaways: Since we adopt a contrastive learning approach and encode the complete hierarchical labels in the vector space without partitioning, the problem of **class imbalance** is tackled. Also, the proposed interactive attention approach captures the **label structure** in the content representations and the chosen sequence representation method for the labels preserves compositionality in hierarchical labels. We also provide theoretical bounds for the approximation involved in the proposed attention mechanism. TagRec++ can also adapt to slight changes in the label space, and we present preliminary analysis in Section V. A detailed evaluation of open-set identification is beyond the scope of this work, and a qualitative analysis is done to demonstrate the advantage of the design choices for the proposed approach. Further, we also demonstrate the **extensibility** of learned representations for tagging learning objectives in a zero-shot setting.

The core contributions of our work are:

- We propose the task of automated content categorization involving hierarchical labels as a dense retrieval problem. We adopt a contrastive learning approach to handle the *class imbalance*.
- To capture the *hierarchical structure* of the labels and similarity to content, we propose an interactive attention approach between the content representations and the hierarchical labels to fuse the information in the hierarchical labels with the input representations for better learning and convergence.
- We render the interactive attention process efficient by grouping related labels to reduce the label space and provide bounds for the approximation, leveraging Lipschitz continuity principle.
- We experimentally study the proposed approach on diverse datasets. We also release a new dataset (KhanAcad) for

learning content categorization. We also demonstrate the *extensibility* of the model to downstream tasks by zero-shot evaluation on learning objectives dataset.

The rest of the paper is organized as follows:

- We present a literature review of text classification methods involving labels of hierarchical structure in Section II.
- We present in detail the proposed approach in Section III. We explain the various components of the proposed approach.
- In Section IV, we describe the data preparation method, experimental setup, baselines and the ablation studies performed.
- We present an analysis of results in Section V.
- We conclude with scope for future work in Section VI.

Reproducibility: We open-source our code and datasets at https://github.com/ADS-AI/TagRec_Plus_Plus_TKDE.

II. Related Work

In this section, we provide an overview of approaches that tackle problems involving labels of hierarchical nature and vector representation methods.

A. Text Categorization to Hierarchical Labels

The online systems use a standardized taxonomy to organize their content [1], [2]. The taxonomy is of hierarchical nature and usually, the approaches used to categorize content in such taxonomy can be categorized into multi-class classification or hierarchical multi-step approaches [11], [12]. In multi-class single-step methods, the leaf nodes are considered labels while ignoring the hierarchy. This leads to class imbalance issue where a large number of samples cover only a few leaf nodes considered as labels. In the hierarchical multi-step approach, the root category is predicted using a classifier and the process is repeated to predict the nodes at the subsequent level. However, the main issues of the approach are that the number of classifiers increases with depth, and the error from one level propagates to the next level. This also increases the computation needed at the inference time. Along similar lines, Banerjee et al. [13] proposed to build a classifier for each level. However, unlike the previous works, the parameters of the classifier for parent levels are transferred to the classifiers at child levels. Another approach [14] proposed to use a chain of neural networks to categorize content to hierarchical labels. A classifier is designated for each level in the hierarchy. However, the major limitation here is that the number of networks in the chain increases with depth, and it also requires that the paths in the label hierarchy should be of the same length, limiting the applications to cases of minor changes in the label space.

To circumvent these issues, each hierarchical taxonomy could be considered as a label disregarding the hierarchy, and a regular single-step classifier could be trained to classify the content to one of the labels. Several single-step classifiers have been proposed for classification tasks involving hierarchical labels. In [12], the word level features like n-grams were used with SVM as a classifier to predict level 1 categories, whereas in [2] the authors have leveraged n-gram features and distributed representations from Word2Vec followed by a linear classifier for multi-class classification. Several deep learning methods like CNN [15] and LSTM [11] have been proposed for the task of question classification. Since the pre-trained language models, like BERT [16], improve the performance, the authors in [1] propose a model BERT-QC, which fine-tunes BERT to categorize questions from the science domain to labels of hierarchical format. The problems involving hierarchical labels have also been formulated as a translation problem in [17] where the product titles are provided as input and use a seq2seq architecture to translate them to product categories having hierarchical structure. The hierarchical neural attention model [18] leverages attention to obtain useful input sentence representation and uses an encoder-decoder architecture to predict each node in the hierarchical learning taxonomy. However, this approach may not scale as the depth of the hierarchy increases.

Several clustering approaches like [19], [20], [21] are also relevant for the considered task as they aid in considering inter-sample similarity for label categorization.

Several works have also tried to capture the hierarchical structure of the labels for the purpose of text categorization to such labels. For instance, Zhou et al. [22] proposed to design an encoder that incorporates prior knowledge of label hierarchy to compute label representations. However, they flatten the hierarchy and treat every label as a leaf node which would require re-training when there are changes in the label space. Lu et al. [23] introduced different types of label graphs (co-occurrence based and semantic similarity based) to improve text categorization. However, they also cast the task of categorizing text to hierarchical labels as a multiple binary classification task [24]. These approaches do not consider the relationship between the terms in the text inputs and the hierarchical labels.

B. Sentence Representation Methods

The NLP tasks like classification and retrieval have been advanced by distributed representations that capture the semantic relationships [25]. Methods like GloVe [26] compute static word embeddings which do not consider the context of occurrence of the word. An unsupervised method named Sent2Vec [27] was proposed to create useful sentence representations.

The Bidirectional Encoder Representation from Transformers (BERT) [16] does not compute any independent sentence representations. To tackle this, Sentence-BERT [28] and Universal Sentence Encoder (USE) [29] models were proposed to generate useful sentence embeddings by fine-tuning transformer based models.

In this paper, sentence representation methods are used to represent the labels by treating each of them as a sequence. For example, the label **Science - Physics - alternating current** is treated as a sequence. We employ sentence representation methods to model the compositionality of terms present in the hierarchical learning taxonomy. Several works [30] [31] have demonstrated that sentence representation models are able to capture the nature of how terms compose together to form meaning in a sequence. We posit that the same principle can be

used to capture the complete semantic meaning of hierarchical taxonomy in the vector space.

Difference between TagRec [10] and TagRec++: In the work TagRec [10], the authors take a dense retrieval approach with the aim to retrieve the relevant label (i.e., the hierarchical learning taxonomy) by aligning the input embeddings and the label embeddings. The authors propose a bi-encoder based architecture [32] [33] to accomplish the task of aligning the inputs with the correct hierarchical labels. Though the input and label representations are matched using a hinge ranking loss, the proposed approach doesn't explicitly combine the information from the input and label spaces. We posit that explicitly capturing the relationship between the terms in the input and the hierarchical labels can help in high recall retrieval. Hence, we propose an interactive attention mechanism where the related labels closer to the input are grouped and attention is computed with respect to all terms in the input to produce hierarchical taxonomy aware representations. In the work TagRec [10], the authors use a random in-batch negative sampling [34] for the ranking loss in the contrastive learning setup. However, in dense retrieval, it is common knowledge [35], [36] that hard negatives aid in high recall retrieval. However, the existing approaches usually leverage a warm start Dense Retrieval model to build a cache of hard negatives and constantly refresh them, adding to computational cost. They also do not refresh the hard negatives based on the parameters of the model being trained. On the other hand, this work proposes to use a dynamic in-batch hard negative sampling approach that provides better negatives as the model parameters are updated in the training loop.

In this work, the multiple classifiers approach is not explored owing to the limitations explained earlier in this section.

III. Methodology

In this section, we describe our method for retrieving relevant hierarchical labels for learning content documents, with each document corresponding to a question-answer pair. The architecture for the proposed approach is as shown in Fig. 1. The proposed approach, *TagRec++* aligns the input representations with the corresponding hierarchical labels representations. It is composed of an interactive cross-attention mechanism that fuses information from the hierarchical labels and the input to render taxonomy aware representations. It also uses an in-batch hard negative sampling approach to pull apart negative labels and bring the input representations closer to the positive label representations.

A. Preliminaries

We first introduce the notation and setup before describing our approach. The input to the approach is a corpus of documents, $C = \{D_1, D_2 \ldots D_n\}$. The documents are tagged with hierarchical labels $O = \{(S_1, Ch_1, T_1), (S_2, Ch_2, T_2) \ldots | S_i > Ch_i > T_i\}$ where S_i (root node), Ch_i and T_i (leaf node) denote subject, chapter, and topic, respectively. The goal here is to learn an input representation that is close in the continuous vector space to the correct label representation. We consider the label (S_i, Ch_i, T_i) as a sequence, $(S_i + Ch_i + T_i)$ and obtain a sentence representation for it using pre-trained models. Since the sentence representation encoder is frozen during training, we can precompute and index the embeddings for the hierarchical labels. This also ensures faster inference. We leverage BERT [16] for obtaining contextualized embeddings followed by a linear layer. The linear layer maps the 768-D representation from BERT to the 1024-D or 512-D vector representation. The novelty of the method lies in the implementation of the attention layer and in-batch hard negative sampling, which are discussed in detail later in the Section.

B. Approach Overview

The steps of the proposed approach can be observed from Algorithm 1. During the *training phase*, the input text is cast to a continuous vector space using a BERT [16] base model. The labels are projected to a continuous vector space using a Sentence-BERT model to capture the composition of terms in the label. We adopt a contrastive learning approach to handle the *class imbalance* issue. To capture the interaction of terms in the content with hierarchical labels, we fuse the information through an attention mechanism between the embeddings of related hierarchical labels and the term embeddings of the content. Then we use a hinge rank-based loss to align the input vector representations with the correct label representations. We also propose an in-batch hard-negative sampling approach to pull apart irrelevant labels for a given input content.

C. Interactive Cross-Attention Module for Taxonomy Aware Input Embeddings

The primary goal of the proposed method is to align the vector representations of the input and the relevant hierarchical taxonomy labels. However, their vector representation sub-spaces may not be close to each other. This renders the alignment of the sub-spaces of the input representations and the corresponding label representations a hard problem. However, the tokens in the labels are related to terms in the content. The alignment approach would benefit from capturing the semantic relatedness between various hierarchical labels and a given content. The final vector representations are computed by fusing the information from hierarchical labels and the input. This would result in representations that are aware of the taxonomy and hence could help in performance improvement of the alignment task. To achieve this, the proposed method first retrieves the label embedding closest to the input representation to capture the interaction between the labels and content which are related to each other (step 4 in Algorithm 1). This will reduce the noise induced by unrelated labels when fusing information from the input and the labels to compute *taxonomy aware* input representations.

$$T_{emb} = f_\theta(D_i)$$
$$O_{emb} \leftarrow g_\theta(O)$$
$$O^i_{emb} \leftarrow \text{top}_1(\cos(T_{emb}, O_{emb}))$$

where f_θ is BERT and g_θ refers to sentence representation methods like Sentence BERT or Universal Sentence Encoder (USE). As this process occurs in the training loop, the closest

Algorithm 1: Tag Recommender.

Input: Training set $T \leftarrow$ docs $\{D_1, ..D_n\}$, labels O of form (Subject-Chapter-Topic)
Output: Set of tags for test set, RO
Training (batch mode)
1: Get input text embeddings, $T_{emb} \leftarrow BERT(D)$
2: Obtain label embeddings, $O_{emb} \leftarrow SENT\_BERT(O)$
3: Index$(labels) \leftarrow O_{emb}$
4: Get label embedding closest to input, $O^i_{emb} \leftarrow \text{top}_1(\cos(T_{emb}, \text{Index}(labels)))$
5: Get top labels close to selected label, $L_r \leftarrow ||O^i_{emb} - index(labels)||_2$
6: Project labels to queries, $Q \leftarrow W^Q * L_r$
7: Project input embeddings to key and values, $K \leftarrow W^K * T_{emb}$ and $V \leftarrow W^V * T_{emb}$
8: Compute modified input embeddings, $T_{\text{new}} \leftarrow \frac{\text{Softmax}(Q * K^T)}{\sqrt{d_k}} * V$
9: hard_neg $\leftarrow$ top_$k(\cos(T_{\text{new}}, \text{Index}(labels)))$ excluding $label$ (adaptive hard-negatives)
10: $L(\text{text}, l) \leftarrow \sum_{j \neq l} \max(0, \delta - \cos(T_{\text{new}}, v(l)) + \cos(T_{\text{new}}, v(j)))$ where $v(j) \in$ hard_neg
11: Fine-tune BERT to minimize $loss$ and align T_{new} and corresponding label representations from O_{emb}
Testing Phase
12: Compute embeddings for test set S using fine-tuned BERT $S_{emb} \leftarrow BERT(S)$
13: $RO \leftarrow sorted(Sim(S_{emb}, O_{emb}))$, gives ranked set of labels
14: **return** Top-k labels from RO

label selected depends on the model parameters. The selection process is thereby dynamic and improves with updates to model parameters. When the model gets better at aligning the input and label representations, it will sample better top-1 hierarchical label closer to the input. Our task is modeled as a dense retrieval approach thereby, our goal is to improve the quality of the first label retrieved.

The step-5 in Algorithm 1 retrieves the top-k labels closest to the selected label. We do not compute attention with respect to all labels in the label space to obviate interference from unrelated labels and reduce the computational complexity of attention. We sample a small set of labels that are related to the label closest to the content representations at a given time in the training loop. This step helps to cluster similar labels which are closer to the input and to each other in the vector space. Also, it reduces the complexity involved in computing attention between the input and label representations. The labels sampled approximate the attention distribution well as shown below, adapting the property of attention from the work [37].

The difference between the attention of inputs (Keys) and labels (Queries) can be bounded by the euclidean distance between the labels.

Statement: Given two label representations O^i_{emb}, O^j_{emb}, the difference between attention can be bounded as by the euclidean distance between the label representations.

To arrive at this result, we first start with the principle of Lipschitz continuity for Softmax. Given two queries Q_i and Q_j,

$$\text{Softmax}(Q_i K^T) - \text{Softmax}(Q_j K^T)||_2) \leq \epsilon ||Q_i K^T - Q_j K^T||_2$$

Let ϕ denote the Softmax operation for the rest of the section.

The Softmax operation has a Lipschitz constant less than 1 [38] which implies ϵ equals 1. Hence the attention approximation is bounded by the euclidean distance between the queries

$$||\phi(Q_i K^T) - \phi(Q_j K^T)||_2 \leq ||Q_i - Q_j||_2 ||K||_2 \quad (1)$$

Since $Q_i \leftarrow W^Q * O^i_{emb}$ we can write the above equation as:

$$||\phi(Q_i K^T) - \phi(Q_j K^T)||_2 \leq ||O^i_{emb} W^Q - O^j_{emb} W^Q||_2 ||K||_2 \quad (2)$$

Since the norm of the weight matrix W^Q is the largest eigenvalue of $((W^Q)^T W^Q)^{1/2}$ we modify the above equation as

$$||\phi(Q_i K^T) - \phi(Q_j K^T)||_2) \leq ||O^i_{emb} - O^j_{emb}||_2 ||W^Q||_2 ||K||_2 \quad (3)$$

By (3), the difference between the attention can be hence bound as the euclidean distance between the label representations.

Following this bound, step-5 of Algorithm 1 samples top-k hierarchical labels (L_r) based on their proximity to the top-1 label in the representation space.

The input representations are projected to (K)ey and (V)alue matrices.

$$K \leftarrow W^K * T_{emb}; \qquad V \leftarrow W^V * T_{emb}$$

where W^K and W^V are learnable weights. The sampled label representations are projected to a (Q)uery matrix

$$Q \leftarrow W^Q * L_r$$

and W^Q is also learnable.

Then we propose an interactive (cross) attention mechanism where the compatibility between the labels (Q) and the inputs (K) are captured in the form of an Attention matrix (A). Then the input representations are weighted by the attention weights in A to promote useful dimensions and drown out irrelevant ones.

$$\alpha = \frac{\text{Softmax}(Q \cdot K^T)}{\sqrt{d_k}}$$

$$T_{\text{new}} \leftarrow \alpha \cdot V$$

where, T_{new} is now the vector representation that fuses the information from content and the representations of the sampled hierarchical labels (Step 8). In the above equation, $Q \in R^{l \times d}$, $K \in R^{n \times d}$ and $V \in R^{n \times d}$. Here l is the number of labels sampled and n is the number of words in the questions (input content). Finally our output embedding from the interactive attention layer $T_{\text{new}} \in R^{l \times d}$ as there are l labels sampled during training.

We finally average across the label dimension to compute a fixed length representation for the given question yielding

$$T_{\text{new}} = \text{mean}(T_{\text{new}}, dim = 0)$$

The computed representations are aligned with the corresponding label representations and pulled apart from representations of negative labels using a hinge rank loss function as explained in Section III-D.

D. Adaptive Hard-Negative Sampling

After the modified input representations are obtained, we proceed to the training step, where the input representations are aligned with the corresponding label representations and pushed apart from the representations of negative labels using a hinge rank loss [39].

For learning representations that disentangle the vector representations of positive and negative labels, we sample hard negatives when optimizing the loss function. The hard negatives are those hierarchical labels with a high semantic relatedness score (cosine similarity) to the input questions but are not the correct hierarchical labels. We sample them using the following equation:

$$\text{hard\_neg} \leftarrow \text{top\_}k(\cos(T_{\text{new}}, \text{Index}(labels)))$$

where $\text{Index}(labels)$ refers to the in-batch hierarchical labels and $label \notin$ hard_neg, $k < batch\_size$. We experiment with different values of k and observe that k =5 gives the best results. After sampling the ground truth hierarchical label as positive and the hard negatives, the hinge rank loss is employed to optimize for the alignment of input and label representations.

The hinge ranking loss is defined as :

$$L(\text{text}, l) \leftarrow \sum_{j \neq l} \max(0, \delta - \cos(T_{emb}, v(l)) + \cos(T_{emb}, v(j)))$$

$$L(\text{text}, l) \leftarrow \frac{L(\text{text}, l)}{len(\text{hard\_neg})}$$

where, $j \in$ hard_neg, T_{emb} denotes the input text embeddings from BERT, $v(label)$ denotes the vector representation of the correct label, $v(j)$ denotes the vector representation of an incorrect label. The margin value was set to 0.1, which is a fraction of the norm of the embedding vectors (1.0), and resulted in the best performance.

The hard negatives are sampled dynamically during the training and hence are a function of model parameters. This implies, hard_neg $\leftarrow f(\theta)$, where $f(\theta)$ denotes the BERT model and θ denotes the model parameters.

At each iteration in the training loop, we sample the incorrect labels which are closer in the vector space to the input representations. This ensures that the hard negatives improve as the model parameters are updated to better align with the correct label representations. We conduct several ablation studies to compare with random negative sampling and demonstrate that the proposed method aids in high recall retrieval.

IV. Experiments

In this section, we discuss the baselines, experimental setup and the datasets used. All experiments are carried out on a tesla T4, 16 GB.

A. Datasets

To evaluate the efficacy of the method *TagRec++*, we perform experiments on the following datasets:

- *QC-Science:* This dataset contains 47832 question-answer pairs belonging to the science domain tagged with hierarchical labels of the form subject - chapter - topic. The dataset was collected with assistance from a leading e-learning platform. The dataset consists of 40895 training samples, 2153 validation samples and 4784 test samples. Some samples are shown in Table I. The average number of words per question is 37.14, and per answer, it is 32.01.
- *ARC [1]:* This dataset consists of 7775 science multiple choice exam questions with answer options and 406 hierarchical labels. The average number of words per question is 20.5. The number of samples in the train, validation, and test sets are 5597, 778 and 1400, respectively.
- *KhanAcad:* We release a new dataset of KhanAcademy video transcripts [1] with corresponding hierarchical labels. This dataset consists of 416 hierarchical labels. The average number of words per question is 822.93. We set maximum length to 512 due to BERT limitations. The number of samples in the train, validation, and test sets are 4188, 924 and 1047, respectively.

In our experiments, for ARC and QC-Science, the question and the answer (QA) are concatenated and used as the input to the model (BERT), and the hierarchical taxonomy is considered as the label. The number of tokens of each QA pair is within 512, within the context limit of BERT.

B. Analysis of Taxonomy Representation Methods

In this section, we provide an analysis of results from a meta-experiment to decide the best sentence representation methods for the hierarchical labels (learning taxonomy). We embed the hierarchical labels using methods like Sent2Vec [27], GloVe and Sentence-BERT [28]. We then compute the semantic similarity between the resulting representations of two different hierarchical labels, as shown in Table II. From Table II, we observe that though "science → physics → electricity" and "science → chemistry → acids" are different, a high similarity score is obtained between representations obtained using GloVe embeddings. This may be due to the observation that averaging word vectors can result in loss of information. Additionally, the context of words like physics is not taken into account when encoding the word electricity. Additionally, the words "physics" and "chemistry" are co-hyponyms which may result in their vectors being close in the continuous vector space when using traditional static embedding methods. We also observe that static sentence embeddings from Sent2Vec are also unable

[1] https://github.com/Khan/khan-api

TABLE I
SOME SAMPLES FROM THE QC-SCIENCE DATASET

Question	Answer	Taxonomy
The value of electron gain enthalpy of chlorine is more than that of fluorine. Give reasons	Fluorine atom is small so electron charge density on F atom is very high	Science→chemistry→classification of elements and periodicity in properties
What are artificial sweetening agents?	The chemical substances which are sweet in taste but do not add any calorie	Science→chemistry→chemistry in everyday life

TABLE II
COMPARISON OF DIFFERENT REPRESENTATION METHODS FOR HIERARCHICAL LABELS

Method	Label1 (L1)	Label2 (L2)	cos(L1, L2)
Sentence-BERT	science → physics → electricity	science → chemistry → acids	0.3072
Sent2vec	science → physics → electricity	science → chemistry → acids	0.6242
GloVe	science → physics → electricity	science → chemistry → acids	0.6632

TABLE III
PERFORMANCE COMPARISON OF TAGREC++ WITH VARIANTS AND BASELINES, † INDICATES TAGREC++'S SIGNIFICANT IMPROVEMENT AT 0.001 LEVEL USING *T-TEST*

Dataset	Method	R@1	R@3	R@5	MRR@1	MRR@3	MRR@5
QC-Science	TagRec++(BERT+USE) (ours)	**0.65** †	**0.84** †	**0.89** †	**0.65** †	**0.74** †	**0.75** †
	TagRec++(BERT+SB) (ours)	**0.65** †	**0.85** †	**0.90** †	**0.65** †	**0.75** †	**0.76** †
	TagRec(BERT+USE) [10]	0.54	0.78	0.86	0.54	0.65	0.67
	TagRec(BERT+SB) [10]	0.53	0.77	0.85	0.53	0.64	0.66
	BERT+sent2vec	0.43	0.70	0.79	0.43	0.56	0.58
	Twin BERT [9]	0.32	0.60	0.72	0.32	0.44	0.47
	BERT+GloVe	0.39	0.65	0.76	0.39	0.50	0.53
	BERT classification (label relation) [1]	0.19	0.33	0.39	0.19	0.25	0.27
	BERT classification (prototypes) [21]	0.54	0.75	0.83	0.54	0.63	0.65
	Pretrained Sent_BERT	0.11	0.22	0.30	0.11	0.16	0.18
	HyperIM [41]	0.57	0.79	0.85	0.57	0.33	0.21
ARC	TagRec++(BERT+USE) (ours)	**0.48** †	**0.66** †	**0.75** †	**0.48** †	**0.56** †	**0.58** †
	TagRec++(BERT+SB) (ours)	**0.49** †	**0.71** †	**0.78** †	**0.49** †	**0.59** †	**0.61** †
	TagRec(BERT+USE) [10]	0.35	0.55	0.67	0.35	0.44	0.47
	TagRec(BERT+SB) [10]	0.36	0.55	0.65	0.36	0.44	0.46
	BERT+sent2vec	0.22	0.43	0.55	0.22	0.28	0.31
	Twin BERT [9]	0.14	0.31	0.46	0.14	0.21	0.24
	BERT+GloVe	0.23	0.43	0.56	0.23	0.32	0.35
	BERT classification (label relation) [1]	0.11	0.21	0.27	0.11	0.15	0.16
	BERT classification (prototypes) [21]	0.35	0.54	0.64	0.35	0.43	0.45
	Pretrained Sent_BERT	0.12	0.24	0.31	0.12	0.17	0.19
	HyperIM [41]	0.20	0.34	0.40	0.20	0.17	0.12
KhanAcad	TagRec++(BERT+USE) (ours)	**0.37** †	**0.50** †	**0.55** †	**0.37** †	**0.43** †	**0.44** †
	TagRec++(BERT+SB) (ours)	**0.38** †	**0.54** †	**0.61** †	**0.38** †	**0.45** †	**0.46** †
	TagRec(BERT+USE) [10]	0.30	0.45	0.50	0.30	0.37	0.38
	TagRec(BERT+SB) [10]	0.26	0.44	0.52	0.26	0.34	0.36
	BERT+sent2vec	0.16	0.32	0.42	0.16	0.23	0.25
	Twin BERT [9]	0.10	0.22	0.32	0.10	0.15	0.17
	BERT+GloVe	0.16	0.31	0.41	0.16	0.22	0.25
	BERT classification (label relation) [1]	0.06	0.14	0.20	0.06	0.09	0.11
	BERT classification (prototypes) [21]	0.18	0.30	0.35	0.18	0.23	0.24
	Pretrained Sent_BERT	0.06	0.11	0.15	0.05	0.08	0.09
	HyperIM [41]	0.19	0.29	0.34	0.19	0.13	0.10

to capture the context of the tokens in the labels, as the representations obtained from Sent2Vec result in a high similarity score. However, we observe that the representations obtained using sentence transformer-based methods like Sentence-BERT are not very similar, as indicated by the similarity score. This indicates that Sentence-BERT is able to produce meaningful sentence representations leveraging the context of tokens for the hierarchical labels.

C. Hyperparameters

We use the BERT base model with 12 encoder layers, 12 attention heads and 768-dimensional output representations in the proposed approach and in other related baselines for fair comparison. We use BERT-base as backbone, following the state-of-the-art approaches like TwinBERT [9], [40]. However, our approach is modular and allows for using other encoder based models too as backbone. We use the AdamW optimizer

TABLE IV
ABLATION ANALYSIS OF TAGREC++

Dataset	Method	R@1	R@3	R@5	MRR@1	MRR@3	MRR@5
QC-Science	TagRec++(BERT+USE) (proposed method)	**0.65**	0.84	0.89	**0.65**	0.74	0.75
	TagRec++(BERT+SB) (proposed method)	**0.65** †	**0.85**†	**0.90**†	**0.65**†	**0.75**†	**0.76**†
	TagRec++(BERT+USE) (- attention)	0.62	0.83	0.88	0.62	0.69	0.70
	TagRec++(BERT+SB) (- attention)	0.62	0.83	0.87	0.62	0.70	0.71
	TagRec++(BERT+USE) (- hard-negatives)	0.57	0.81	0.86	0.57	0.69	0.70
	TagRec++(BERT+SB) (- hard-negatives)	0.56	0.80	0.87	0.56	0.64	0.66
ARC	TagRec++(BERT+USE) (proposed method)	0.48	0.69	0.75	**0.48**	0.56	0.58
	TagRec++(BERT+SB) (proposed method)	**0.49**†	**0.71**†	**0.77**†	**0.48**†	**0.59**†	**0.61**†
	TagRec++(BERT+USE) (- attention)	0.41	0.61	0.70	0.41	0.47	0.49
	TagRec++(BERT+SB) (- attention)	0.44	0.64	0.74	0.44	0.52	0.54
	TagRec++(BERT+USE) (- hard-negatives)	0.39	0.60	0.72	0.39	0.48	0.51
	TagRec++(BERT+SB) (- hard-negatives)	0.45	0.66	0.74	0.45	0.54	0.56
KhanAcad	TagRec++(BERT+USE) (ours)	0.37	0.50	0.55	0.37	0.43	0.44
	TagRec++(BERT+SB) (ours)	**0.38** †	**0.54** †	**0.61** †	**0.38** †	**0.45** †	**0.46** †
	TagRec++(BERT+USE) (- attention)	0.33	0.49	0.53	0.33	0.40	0.41
	TagRec++(BERT+SB) (- attention)	0.26	0.44	0.53	0.26	0.34	0.36
	TagRec++(BERT+USE) (- hard-negatives)	0.32	0.48	0.53	0.32	0.39	0.41
	TagRec++(BERT+SB) (- hard-negatives)	0.31	0.48	0.57	0.31	0.39	0.41

TABLE V
EXAMPLES DEMONSTRATING THE PERFORMANCE FOR UNSEEN LABELS AT TEST TIME

Question text	Ground truth	Top 2 predictions	Method
A boy can see his face when he looks into a calm pond. Which physical property of the pond makes this happen? (A) flexibility (B) reflectiveness (C) temperature (D) volume	matter→properties of material→reflect	matter→properties of material→flex and **matter→properties of material→reflect**	TagRec++ (BERT+USE)
		matter→properties of objects→mass and matter→properties of objects→density	Twin BERT [9]
		matter→states→solid and matter→properties of material→density	BERT+GloVe
		matter→properties of material→specific heat and matter→properties of material	BERT+sent2vec
Which object best reflects light? (A) gray door (B) white floor (C) black sweater (D) brown carpet	matter→ properties of material→reflect	energy→light→reflect and **matter→properties of material→reflect**	TagRec++ (BERT+USE)
		energy→thermal→ radiation and energy→light→generic properties	Twin BERT [9]
		energy→light and energy→light→refract	BERT+GloVe
		energy→light→reflect and energy→light→refract	BERT+sent2vec

TABLE VI
PERFORMANCE COMPARISON FOR ZERO-SHOT LEARNING OBJECTIVE CATEGORIZATION

Method	R@1
TagRec++(BERT+SB) (ours)	**0.82**
TagRec++(BERT+USE) (ours)	0.79
TagRec(BERT+USE) [10]	0.69
TagRec(BERT+SB)	0.77
BERT+sent2vec	0.49
Twin BERT [9]	0.54
BERT+GloVe	0.62
BERT classification (label relation) [1]	0.46
BERT classification (prototypes) [21]	0.60
Pretrained Sent_BERT	0.39

Note: here R@1 is same as MRR@1.

with learning rate of 2e-5 for training the models. The batch size was set to 32 and all models including baselines were trained for 30 epochs with early stopping with a patience value of 6. All hyperparameters mentioned were finalized using the validation set across all datasets. The same process was followed to finalize hyperparameters for baselines. For the margin parameter δ, in the hinge ranking loss, we experiment with values of [0.1,0.2,0.4,0.6,0.8,1..] and observe 0.1 to be the best.

D. Methods and Experimental Setup

We compare TagRec with flat multi-class classification methods and other state-of-the-art interaction based or contrastive

TABLE VII
ABLATION RESULTS FOR ZERO-SHOT EVALUATION ON LEARNING OBJECTIVE CATEGORIZATION

Method	R@1
TagRec++(BERT+SB) (ours)	**0.82**
TagRec++(BERT+SB) (-attention)	0.80
TagRec++(BERT+SB) (-hard-negatives)	0.80

R@1 is same as MRR@1

learning methods. In TagRec, the labels are represented using transformer based sentence representation methods like Sentence-BERT (Sent_BERT) [28] or Universal Sentence Encoder [29].

The methods we compare against are:

- *BERT+Sent2Vec:* In this method, the training and testing phases are similar to TagRec. The label representations are obtained using Sent2vec [27] instead of USE or Sent_BERT.
- *BERT+GloVE:* In this method, the labels are represented as the average of the word embeddings of their constituent words. The word embeddings are obtained from GloVe. The training and testing phases are the same as TagRec.
- *Twin BERT:* This method is reproduced from the work Twin BERT [9]. In this method, a pre-trained BERT model is fine-tuned to represent the labels in a continuous vector space. The label representations correspond to the first token of the last layer hidden state, denoted as [CLS] in BERT.
- *BERT multi-class (label relation) [1]:* In this method, the hierarchical labels are flattened and encoded, resulting in a multi-class classification method. Then we fine-tune a pre-trained BERT model for categorizing the input content to the labels. During inference, the representations for the inputs and labels are computed using the fine-tuned model. Then the top-k labels are retrieved based on similarity to input.
- *BERT multi-class (prototypes) [21]:* To provide a fair comparison with *TagRec++*, we propose another baseline that considers the inter-sample similarity. A BERT model is fine-tuned like the previous baseline. Then for each class, we compute the mean of the embeddings of random samples from the training set to serve as the prototype for the class. The vector representation for each selected sample is obtained by the concatenation of the [CLS] token obtained from the last 4 layers of the fine-tuned model. This method of vector representation gives the best performance. The class prototypes are used to retrieve top-k labels.
- *Pretrained Sent_BERT:* We implement a baseline where the input texts and labels are encoded using a pre-trained Sentence-BERT model. Then top-k similar labels are retrieved.
- *TagRec [10]:* We compare with the recent state-of-the-art approach that cast hierarchical taxonomy tagging as a contrastive learning problem.
- *HyperIM:* We also compare with the recent SOTA approach HyperIM [41] which casts the input and the hierarchical label to the hyperbolic space. Then the distance between the input and all label representations are used as a feature vector for classification. This approach cannot adapt to changes in label space.

E. Metrics

We compare the proposed approaches with baselines and state-of-the-art methods using standard IR metrics like Recall@k (R@k) and MRR@k [42], [43] which are popular for retrieval tasks with only one relevant label. Since each sample in our datasets are tagged only with one relevant hierarchical path, R@k helps evaluate if the correct path is among the top-k retrieved learning taxonomy paths. Additionally, MRR (Mean Reciprocal Rank) [44] helps measure if the relevant label is assigned a higher rank among retrieved labels. This is of paramount importance as higher the rank of the relevant label, higher the retrieval quality. Since we have only one relevant label, MRR is also equivalent to Mean Average Precision [44].

V. RESULTS AND ANALYSIS

A. Performance Comparison With Other Approaches

The performance comparison of TagRec++ with baselines and other variants can be observed from Table III. We observe that the TagRec++ method outperforms the flat multi-class classification based baselines, confirming the hypothesis that capturing the semantic relatedness between the terms in the input and tokens in the hierarchical labels results in better representations. This is pivotal to the question-answer pair categorization task as the technical terms in the short input text are semantically related to the tokens in the label. The baseline (BERT label relation) performs poorly as it has not been explicitly trained to align the input and the hierarchical label representations. The representations obtained through the flat multi-class classification approach have no notion of semantic relatedness between the content and label representations. But the prototypical embeddings baseline performs better as the classification is done based on semantic matching between train and test sample representations. However, this baseline also has no notion of semantic relatedness between the input and label representations. Hence, it does not perform well when compared to our proposed method, TagRec++. Moreover, this baseline cannot also adapt to changes in the label space and requires a change in the final classification layer and *retraining*. We also observe that the baseline of semantic matching using pre-trained sentence BERT does not work well.

We observe that contextualized embedding methods for labels provide the best performance. This is evident from the table, as TagRec++(BERT+USE) and TagRec++(BERT+SB) outperform approaches which leverage static sentence embedding methods like BERT+Sent2Vec and BERT+GloVe. This is because transformer-based encoding methods use self-attention to produce better representations. In addition, the Sentence-BERT and the Universal Sentence Encoder models are ideal for retrieval based tasks as they were pre-trained on semantic text similarity (STS) tasks. Also, as measured by **MRR@k**, we observe that the proposed approach *TagRec++* provides a

TABLE VIII
PERFORMANCE COMPARISON (R@K) FOR EACH EPOCH ON ARC DATASET: TAGREC++ VS TAGREC++ (-HARD-NEGATIVES

	Epochs				
Method	**2**	4	6	8	10
TagRec++(BERT+SB) (ours)	**0.30**	**0.40**	**0.43**	**0.45**	**0.45**
TagRec++(BERT+SB) (-hard-negatives)	0.22	0.33	0.35	0.39	0.39

TABLE IX
QUALITATIVE ANALYSIS OF TOP-3 HARD-NEGATIVES SAMPLED ON QC-SCIENCE DATASET

Question	Answer	Epoch	Hard negatives		
			1	2	3
In the given transistor circuit, the base current is 35 μA. The value of R b is	200 Omega	1	science → physics → work, energy and power	physics → part - ii → mechanical properties of fluids	physics → part - ii → thermal properties of matter
		5	science → physics → communication systems	physics → part - i → magnetism and matter	physics → part - i → system of particles and rotational motion
		10	physics → part i → magnetic effects of current	physics → part - i → magnetism and matter	physics → part - i → moving charges and magnetism

TABLE X
PERFORMANCE FOR DIFFERENT VALUES OF K WHEN SAMPLING TOP-K TAXONOMY TAGS FOR ATTENTION IN TAGREC++

# of tags for attention	R@1	R@3	R@5
1	0.61	0.82	0.88
5	**0.65**	**0.85**	**0.90**
10	0.63	0.85	0.89

higher rank to the relevant label compared to other approaches, indicating high retrieval quality.

We perform **statistical significance** tests and observe that the predicted results are statistically significant over *TagRec* at 0.001 level for all 3 datasets. For instance, for Recall@5 we observe that TagRec++ is statistically significant (*t-test*) with p-values **0.0000499** and **0.0000244** for *QC-Science* and *ARC* respectively.

B. Ablation Studies

We also perform several ablation analyses of the proposed TagRec++ approach. As observed in Table IV we compare TagRec++ with and without the proposed interactive attention mechanism. We see a clear performance difference, confirming the hypothesis that the interactive attention mechanism is crucial for high recall retrieval as it captures the relatedness between the tokens in the hierarchical labels and the terms in the input content (questions).

We also performed another ablation study to ascertain the effectiveness of the proposed in-batch hard negative sampling. Instead of sampling in-batch hard negatives, we sample random negatives. The random negatives are also sampled dynamically for a fair comparison. From Table IV, we can observe that the proposed in-batch hard negative sampling works better than the random negative sampling.

Zero-Shot Performance: We curated a set of learning objectives from K-12 textbooks to test the ability of *TagRec++* to tag related short learning content without training. This experiment is performed to observe the zero-shot abilities of *TagRec++*. We observe that *TagRec++* outperforms existing approaches as measured by Recall@k as shown in Table VI. This demonstrates that the proposed approach also leads to high recall retrieval in a zero-shot setting. We also perform certain ablation studies for the zero-shot setting by removing the interactive attention component and the in-batch hard negatives sampling approach as shown in Table VII. We observe that *TagRec++* achieves the highest performance, indicating the significance of the proposed attention mechanism and hard negative sampling method.

Additionally, we observe that TagRec++ was also able to adapt to changes in the label space. For instance, in the *ARC* dataset, two samples in the test set were tagged with *"matter → properties of material → reflect"* unseen during the training phase as shown in Table V. At test time, the label *"matter → properties of material → reflect"* appeared in top 2 predictions output by the proposed method (TagRec++ (BERT + USE)) for the two samples. We observe that for other baselines shown in Table V the correct label does not get retrieved even in top-10 results. Similar results are observed for other baselines and are not shown in Table V owing to space constraints. We only provide preliminary results here and to confirm the hypothesis a detailed comparison is required, which is beyond the scope of this work.

C. Qualitative Analysis

We analyze the hard-negatives sampled in the training loop to determine if the quality of hard negatives increases as training progresses. The in-batch dynamic negative sampling is based on the hypothesis that the model improves with training and samples better hard negatives, leading to high recall retrieval when compared to dynamic random negatives sampling. To test this hypothesis, apart from the ablation shown in Table IV, we also perform an epochwise comparison of dynamic hard negatives and dynamic random negatives as shown in Table VIII. We observe that *TagRec++* with dynamic hard negatives has

TABLE XI
QUALITATIVE ANALYSIS OF TOP-3 TAGS SAMPLED ON QC-SCIENCE DATASET FOR THE CROSS-ATTENTION MECHANISM

Question	Answer	Epoch	hierarchical tags for attention		
			1	2	3
In the given transistor circuit, the base current is 35 μA. The value of R b is	200 Omega	1	physics → part - i → magnetism and matter	physics → part - i → moving charges and magnetism	science →physics→magnetic effects of electric current
		5	physics → part - i → electromagnetic waves	physics → part - i → alternating current	physics → part - i → electrostatic potential and capacitance
		10	physics → part - i → alternating current	physics → part - i → current electricity	physics → part - i → electric current and it's effects

TABLE XII
EXECUTION TIME, WHERE M-MINUTES AND S-SECONDS

Method	Training	Inference
TagRec++ (BERT+SB)	**252 m**	2m20 s
TagRec (BERT+SB)	420 m	2m17 s
Twin BERT	930 m	2m28 s
HyperIM	480m10 s	4m40 s

a higher recall in each epoch due to better sampling when compared to dynamic random negatives.

We also perform a qualitative analysis of the negatives sampled. The observation for a sample is shown in Table IX. We observe in epoch one, the hard negatives are centered around *physics* subject, but the topics are not related to the input. The ground truth label for the question shown in the table is centered around *electrical circuits*. We observe that as training progresses, the top-2 labels are centered around *magnetism*, *communication systems* but do not correspond to the correct theme of *electrical circuits*, rendering them as hard negatives. This demonstrates that dynamic sampling of in-batch hard-negatives improves with training. We observe a similar phenomenon for other samples too, which are not attached here due to space constraints.

We also vary the value of k for the top-k tags (hierarchical labels) sampled for cross-attention, discussed in Section III-A. The results are shown in Table X. We observe that the highest R@k is achieved for a value of 5. When only one tag is sampled, it contains very less information, as demonstrated by the values of R@k. We also observe sampling ten tags to fuse the taxonomy information with the input leads to a lower R@k. This maybe due to noise induced by the tags less related to the input. Hence, we set k in top-k tags sampled to 5. We also perform the qualitative analysis of tags sampled as shown in Table XI. We observe that though in the first epoch, we get tags related to *magnetism* as the training progresses, in later epochs, we get most tags relevant to *electricity* which are closer to the ground truth label *semiconductors and electrical circuits*.

D. Execution Time

We observe that inference is faster in general for dual-encoder models as shown in Table XII, as the embeddings for labels are pre-computed and indexed. We observe that for QC-Science for a test set of 4784 questions the inference time on T4 GPU without batched inference is **2 minute 20 seconds** which is only slightly more than TagRec, due to the efficient cross attention layer but provides huge performance gains. Our approach is also faster than the state.of-the-art hyperbolic space based method, HyperIM. We observe that the proposed *TagRec++* converges faster than other competitive methods owing to the proposed cross-interaction approach.

VI. CONCLUSION AND FUTURE WORK

We proposed a novel approach, TagRec++ for tagging content to hierarchical taxonomy. *TagRec++* incorporates an adaptive in-batch hard-negative sampling approach for achieving high recall retrieval. Further, *TagRec++* uses a cross-attention approach, fusing the information from the content and the hierarchical representations. We observe that the proposed approach outperforms *TagRec*, other baselines. In the future, we plan to embed the hierarchical labels in the hyperbolic space and also study in detail the ability of the approach to tackle changes in label space.

REFERENCES

[1] D. Xu et al., "Multi-class hierarchical question classification for multiple choice science exams," in *Proc. 12th Lang. Resour. Eval. Conf.*, Marseille, France, 2020, pp. 5370–5382.
[2] Z. Kozareva, "Everyone likes shopping! multi-class product categorization for e-commerce," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Hum. Lang. Technol.*, 2015, pp. 1329–1333.
[3] Y. Yang and X. Liu, "A re-examination of text categorization methods," in *Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, New York, NY, USA, 1999, pp. 42–49. [Online]. Available: https://doi.org/10.1145/312624.312647
[4] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proc. 14th Int. Conf. Mach. Learn.*, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., 1997, pp. 412–420.
[5] X. Qiu, J. Zhou, and X. Huang, "An effective feature selection method for text categorization," in *Proc. 15th Pacific-Asia Conf. Adv. Knowl. Discov. Data Mining*, Berlin, Heidelberg, Springer-Verlag, 2011, pp. 50–61.
[6] Y. Yang and Z. Xu, "Rethinking the value of labels for improving class-imbalanced learning," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, Red Hook, NY, USA, 2020, pp. 19290–19301.
[7] Y. Marrakchi, O. Makansi, and T. Brox, "Fighting class imbalance with contrastive learning," in *Proc. Med. Image Comput. Comput. Assist. Intervention*, 2021, pp. 466–476.
[8] S. Zhang, R. Xu, C. Xiong, and C. Ramaiah, "Use all the labels: A hierarchical multi-label contrastive learning framework," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16639–16648, doi: 10.1109/CVPR52688.2022.01616.

[9] W. Lu, J. Jiao, and R. Zhang, "Twinbert: Distilling knowledge to twin-structured compressed bert models for large-scale retrieval," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 2645–2652.

[10] V. Venktesh, M. Mohania, and V. Goyal, "TagRec: Automated tagging of questions with hierarchical learning taxonomy," 2021. [Online]. Available: https://arxiv.org/abs/2107.10649

[11] W. Xia, W. Zhu, B. Liao, M. Chen, L. Cai, and L. Huang, "Novel architecture for long short-term memory used in question classification," *Neurocomputing*, vol. 299, pp. 20–31, 2018.

[12] H.-F. Yu, C.-H. Ho, P. Arunachalam, M. Somaiya, and C.-J. Lin, "Product title classification versus text classification," Tech. Rep., Dept. Comput. Sci., Nat. Taiwan Univ., Taipei, Taiwan.

[13] S. Banerjee, C. Akkaya, F. Perez-Sorrosal, and K. Tsioutsiouliklis, "Hierarchical transfer learning for multi-label text classification," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Florence, Italy, 2019, pp. 6295–6300. [Online]. Available: https://aclanthology.org/P19--1633

[14] J. Wehrmann, R. C. Barros, S. N. D. Dôres, and R. Cerri, "Hierarchical multi-label classification with chained neural networks," in *Proc. Symp. Appl. Comput.*, New York, NY, USA, 2017, pp. 790–795. [Online]. Available: https://doi.org/10.1145/3019612.3019664

[15] T. Lei, Z. Shi, D. Liu, L. Yang, and F. Zhu, "A novel CNN-based method for question classification in intelligent question answering," in *Proc. Int. Conf. Algorithms Comput. Artif. Intell.*, New York, NY, USA, 2018, pp. 1–6.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics - Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[17] L. Tan, M. Y. Li, and S. Kok, "E-commerce product categorization via machine translation," *ACM Trans. Manage. Inf. Syst.*, vol. 11, no. 3, pp. 1–14, 2020.

[18] K. Sinha, Y. Dong, J. C. K. Cheung, and D. Ruths, "A hierarchical neural attention-based text classifier," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 817–823.

[19] Z. Li, F. Nie, X. Chang, Y. Yang, C. Zhang, and N. Sebe, "Dynamic affinity graph construction for spectral clustering using multiple features," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6323–6332, Dec. 2018.

[20] Z. Li, F. Nie, X. Chang, L. Nie, H. Zhang, and Y. Yang, "Rank-constrained spectral clustering with flexible embedding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6073–6082, Dec. 2018.

[21] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4077–4087.

[22] J. Zhou et al., "Hierarchy-aware global model for hierarchical text classification," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 1106–1117. [Online]. Available: https://aclanthology.org/2020.acl-main.104

[23] J. Lu, L. Du, M. Liu, and J. Dipnall, "Multi-label few/zero-shot learning with knowledge aggregated from multiple label graphs," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 2935–2943. [Online]. Available: https://aclanthology.org/2020.emnlp-main.235

[24] J. Fürnkranz, E. Hüllermeier, E. L. Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Mach. Learn.*, vol. 73, pp. 133–153, 2008.

[25] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, Red Hook, NY, USA, Curran Associates Inc., 2013, pp. 3111–3119.

[26] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Doha, Qatar, 2014, pp. 1532–1543.

[27] M. Pagliardini, P. Gupta, and M. Jaggi, "Unsupervised learning of sentence embeddings using compositional n-gram features," 2017, *arXiv: 1703.02507*.

[28] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process.*, Hong Kong, China, 2019, pp. 3982–3992.

[29] D. Cer et al., "Universal sentence encoder for English," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, Brussels, Belgium, 2018, pp. 169–174.

[30] H. Bhathena, A. Willis, and N. Dass, "Evaluating compositionality of sentence representation models," in *Proc. 5th Workshop Representation Learn. NLP*, 2020, pp. 185–193. [Online]. Available: https://aclanthology.org/2020.repl4nlp-1.22

[31] A. Ettinger, A. Elgohary, C. Phillips, and P. Resnik, "Assessing composition in sentence vector representations," in *Proc. 27th Int. Conf. Comput. Linguistics*, Santa Fe, New Mexico, USA, 2018, pp. 1790–1801. [Online]. Available: https://aclanthology.org/C18--1152

[32] P.-E. Mazaré, S. Humeau, M. Raison, and A. Bordes, "Training millions of personalized dialogue agents," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Brussels, Belgium, 2018, pp. 2775–2779. [Online]. Available: https://aclanthology.org/D18--1298

[33] E. Dinan et al., "The second conversational intelligence challenge (convai2)," 2019. [Online]. Available: https://arxiv.org/abs/1902.00098

[34] J.-T. Huang et al., "Embedding-based retrieval in facebook search," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, New York, NY, USA, 2020, pp. 2553–2561. [Online]. Available: https://doi.org/10.1145/3394486.3403305

[35] L. Xiong et al., "Approximate nearest neighbor negative contrastive learning for dense text retrieval," 2020. [Online]. Available: https://arxiv.org/abs/2007.00808

[36] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "Realm: Retrieval-augmented language model pre-training," JMLR.org, 2020, Art. no. 368.

[37] A. Vyas, A. Katharopoulos, and F. Fleuret, "Fast transformers with clustered attention," 2020. [Online]. Available: https://arxiv.org/abs/2007.04825

[38] B. Gao and L. Pavel, "On the properties of the softmax function with application in game theory and reinforcement learning," 2017. [Online]. Available: https://arxiv.org/abs/1704.00805

[39] A. Frome et al., "Devise: A deep visual-semantic embedding model," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2121–2129.

[40] R. Nogueira, W. Yang, K. Cho, and J. Lin, Multi-stage document ranking with BERT, 2019, *arXiv:1910.14424*.

[41] B. Chen, X. Huang, L. Xiao, Z. Cai, and L. Jing, "Hyperbolic interaction model for hierarchical multi-label classification," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 7496–7503. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/6247

[42] V. Bhatnagar, D. Kanojia, and K. Chebrolu, "Harnessing abstractive summarization for fact-checked claim detection," in *Proc. 29th Int. Conf. Comput. Linguistics*, Gyeongju, Republic of Korea, 2022, pp. 2934–2945. [Online]. Available: https://aclanthology.org/2022.coling-1.259

[43] Y. Wang et al., "A neural corpus indexer for document retrieval," *Adv. Neural Inf. Process. Syst.*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=fSfcEYQP_qc

[44] N. Craswell, *Mean Reciprocal Rank*, Boston, MA: Springer, 2009, pp. 1703–1703. [Online]. Available: https://doi.org/10.1007/978--0-387-39940-9_488

Venktesh V received the PhD degree from the Department of Computer Science and Engineering, IIIT-Delhi, in 2023. He is currently a postdoctoral researcher with TU DELFT. He is also a PM fellow with SERB-FICCI. His research interests include neural information retrieval and natural language processing.

Mukesh Mohania is a professor with the Department of Computer Science and Engineering and Dean of Innovation, Research and Development, IIIT-Delhi. His research interests are on Information (structured and unstructured data) integration, natural language processing, AI based entity analytics, and Big Data analytics and applications.

Vikram Goyal is a professor with the Department of Computer Science and Engineering in IIIT-Delhi. He has many publications in reputed conferences and referred journals. He is the director of TiH Anubhuti (IIITD) and program director of PG Diploma in Data Science and AI. He is also a member of Infosys Centre for AI, IIITD. His research interests are in information retrieval, data mining, databases, and spatial data analytics