Weakly-Supervised Hierarchical Multi-Label Text Classification by Interacting Multi-Granular Text Units

Seonggeun Cho
University of Illinois at Urbana-Champaign
Champaign, USA
sc27@illinois.edu

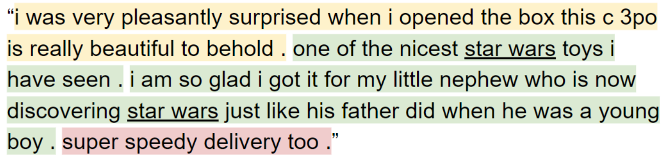
Yuhong Shao
University of Illinois at Urbana-Champaign
Champaign, USA
yuhongs2@illinois.edu

***Abstract*—This project proposes a weakly-supervised hierarchical multi-label text classification (HMTC) approach that exploits the potential of multi-granular text units for document classification. Central to our approach is the hypothesis that leveraging sentence-level information can augment the effectiveness of document-level classification. To realize this, we formulate a multi-granular graph representation characterized by three primary nodes: documents, sentences, and keywords. Addressing the intricacies of out-of-vocabulary surface names, SeeTopic is employed to pinpoint class-indicative keywords, facilitating keyword matching. For the sentence nodes, pseudo labels are generated using PLM entailment model. Using these annotations, we conduct iterative contrastive learning within the graph. The graph undergoes refinement by propagating pseudo label data and regulating its structure by diminishing the relevance of extraneous nodes with multi-head attention mechanism. Concurrently, representation learning on the graph is pursued with contrastive learning, and document-level representation is formed by aggregating sentence-level embeddings. Due to the inherent problems with the current proposed approach, experiments on the Amazon-531 dataset yielded unsatisfactory results. Therefore, as a future direction, we would like to improve the current methodology by enhancing the quality of pseudo-labels and handling the dataset imbalance problem.**

1. Introduction

The field of Natural Language Processing (NLP) underscores the paramount significance of text classification owing to its diverse applications and far-reaching implications. This fundamental process involves categorizing documents into predefined classes, serving as a cornerstone for extracting meaningful insights from extensive textual datasets. The development of big data analysis has introduced more diverse and complex textual contents into the picture, necessitating a more sophisticated classification framework.

Hierarchical Multi-Label Text Classification (HTMC) emerges as a pivotal solution in this context. It encompasses the task of categorizing text documents within a predetermined hierarchical label taxonomy. HTMC is positioned to adeptly accommodate a broad spectrum of topics and subtopics, harnessing the hierarchical information inherent in its structure, thus leading to better performance. The ramifications of HTMC extend across diverse domains, finding application in scientific document categorization, medical diagnosis, content tagging in social media, and product classification.



"i was very pleasantly surprised when i opened the box this c 3po is really beautiful to behold . one of the nicest star wars toys i have seen . i am so glad i got it for my little nephew who is now discovering star wars just like his father did when he was a young boy . super speedy delivery too ."

Figure 1: Example of Multi-Granular Text

In recent times, the field of weakly supervised text classification has garnered noteworthy attention [8]. Operating in conditions where human-labeled training data is scarce and label-specific information is limited, weakly supervised text classification stands in contrast to traditional supervised classifier training, offering a cost-effective alternative that requires minimal information.

However, existing efforts have predominantly focused on direct document classification, inadvertently overlooking the rich information encapsulated within sentences and tokens [6]. Illustrated in Figure 1, in the context of a product review, sentences within the same review may exhibit semantic differences, with some directly related to the product and others detailing the general delivery procedure. Recognizing that finer-grained label distinctions within a hierarchical taxonomy can be elucidated more effectively through sentence-level and token-level information, our focus centers on addressing the challenge of hierarchical text classification. In this pursuit, we introduce an innovative weakly supervised HMTC framework, adept at capturing multiple granularities through a multi-graph structure and deriving document label aggregations tailored to classifiers.

In summary, this paper presents the following noteworthy contributions:

1) We introduce an enhanced multi-granular graph structure, which encapsulates embedding information at document-level, sentence-level, and word-

level, with provisions for meticulous regulation and refinement through representation learning.

2) We formulate a comprehensive training strategy that thoughtfully incorporates the category taxonomy when undertaking classification tasks.

2. Related Work

Weakly-supervised NLP Tasks In the realm of weakly supervised learning, strategies for augmenting limited seed information typically involve two prominent approaches: enrichment via corpus specificity and enrichment through pre-trained language models (PLM). For instance, JoSH introduces the task of hierarchical topic mining, aiming to extract a set of representative keywords from a text corpus based on a user-defined class hierarchy [4]. This is achieved through the joint learning of text embeddings and tree embeddings in the spherical space, emphasizing the maximization of directional similarity to characterize semantic correlations. In a similar vein, SeeTopic further addresses the challenge of out-of-topic seed words by employing a PLM to generate initial representations, followed by the refinement of location representations using point-wise mutual information as an objective function [11].

Hierarchical Text Classification When conducting classification based on a user-defined class hierarchy, the challenges lies in developing a strategy to incorporate hierarchical information into the training process. WeSH Class [2] learns a hierarchy of training instances and optimizes a ranking-based objective at each node of the hierarchy through iterative self-training using confident predictions. [6] introduced a unique framework for HTMC. Using only class names as supervisionary signals, it bypasses the need for extensive human-labeled datasets. The process imitates human experts by identifying core classes and examining their ancestral classes.

Multi-level granularities MEGClass [1] employs the synergy between varying text granularities, such as documents, sentences, and words. By leveraging keyword-based sentence representations, it computes primary class distributions for documents, providing a foundation for a multi-head attention network. On the other hand, the complex and diverse nature of text granularities motivates the use of graphical Neural Networks (GNN). Using a GNN, more complex structural information and dependencies can be captured within the deep learning model. For example, ClassKG [8] addresses the shortcomings of keyword-driven methods by tapping into keyword correlations via GNN. It iteratively transforms pseudo label assignments into keyword subgraph annotations, refining the process with a self-supervised task. Finally, FuTex [9] emphasized the structural signals of scientific papers. Using network-aware contrastive fine-tuning, it captures fine-grained label semantics through a cross-paper network. Simultaneously, hierarchy-aware aggregation considers the inherent structure within papers, creating a comprehensive document representation.

3. Problem Definition

3.1. Notations

Our research focuses on a weakly-supervised hierarchical multi-label text classification, which involves with processing a corpus and a class taxonomy. Specifically, a corpus is a set of documents, $\mathcal{D} = \{D_1, ..., D_N\}$, where each document $D_i \in \mathcal{D}$ is a sequence of words. A class taxonomy is a directed acyclic graph represented as $\mathcal{T} = (\mathcal{C}, \mathcal{R})$, where each node represents a class c_j , and each directed edge $< c_m, c_n > \in \mathcal{R}$ indicates a hierarchical relationship between the parent class c_m and its child class c_n.

In our approach, documents are subdivided to create a hierarchical document graph, denoted as $\mathcal{G} = \{G_1, ..., G_N\}$. This graph is heterogeneous, comprising three distinct types of nodes: document node, sentence nodes, and keyword nodes. Specifically, document node, represented as D_i, where i is the index of the document in the collection. Sentence node is denoted as S_i, where i represents the index of the individual sentence within a document. Each S_i is labeled with zero or more pseudo-labels, $\{c_i, ..., c_N\} \in \mathcal{C}$. Keyword node is indicated as K_i, with i indexing each keyword in the graph. Every K_i is associated with a single pseudo-label $c_i \in \mathcal{C}$.

3.2. Task Definition

Given an unlabeled corpus $\mathcal{D}$, a class hierarchy $\mathcal{T} = (\mathcal{C}, \mathcal{R})$, and a set of hierarchical document graphs $\mathcal{G} = \{G_i\}_{i=1}^{\mathcal{D}}$, our objetive is two-fold:

1. **Learning Multi-Granular Graph Representations**: This task involves with understanding the hierarchically structured graphical information from $\mathcal{G}$ and enhancing the node representation of each level. This is crucial for capturing complex relationship among different granular-level information within the document corpus $\mathcal{D}$.

2. **Developing a Sentence Classifier**, $f(\cdot)$: This task is to formulate a sentence classifier $f(\cdot)$ that maps each sentence node S_i to its target label vector $\mathbf{y} = [y_1, ..., y_{|\mathcal{C}|}]$, where each y_j is 1 if S_i belongs to class c_j, and 0 otherwise. Then we aggregate sentence-level representations to form document-level representations to classify each document in the corpus into corresponding categories, as defined by the class hierarchy $\mathcal{T}$.

4. Methodology

Our proposed framework is structured into five major steps: (1)pseudo-label generation, (2) multi-granular graph representation, (3) contrastive learning on multi-granular graph (4) pseudo-label guided sentence classifier training, and (5) multi-label self-training. Fig.2 provides a visual overview of our framework and details on each step will be discussed in the following sections.

4.1. Pseudo-label Generation

Given that the sole available information regarding labels consists of class surface names, which offer limited insights, our primary aim is to enrich the hierarchical label information. To this end, we make modifications to two existing methods and leverage the power of two different pretrained language models in hope that pseudo label at different text granular levels may complement each other.

4.1.1. Sentence Pseudo Label Generation. For sentence-level pseudo labels, we adopt the strategy in TaxoClass [6], where we first calculate sentence-class similarity score using BART-large-MNLI, and then select confident core classes along the label hierarchy. Finally we filter top 10% confident labels. Details of the algorithm is shown in Algorithm 1.

Algorithm 1 Confident Core Class Selection

```
procedure GENERATE_SENTENCE_PSEUDO_LABEL
    for each sentence s do
        for each level l do
            cani_classes$_{l,s}$ ← child(cani_classes$_{l-1,s}$)
            for cani_class i in cani_classes$_{l,s}$ do
                sim$_{i,s}$ ← entailment_PLM(i, s)
            end for
            Filter cani_classes$_l$ using similarity score.
        end for
    end for
    for each level l do
        Filter candidate classes using path score.
    end for
    for each sentence s do
        for cani_class i in cani_classes$_s$ do
            conf$_{i,s}$ ← confidence(i, neighbour(i))
        end for
    end for
    for each sentence s do
        for cani_class i in cani_classes$_s$ do
            if some condition is true then
                drop(cani_classes$_s$, i)
            end if
        end for
    end for
end procedure
```

4.1.2. Keyword Pseudo Label Generation. For keyword pseudo labels, we adopt the strategy in SeeTopic [11]. However, since SeeTopic generates mutually exclusive keyword sets while in our case child labels belong to parent labels, we extract keywords from differently levels separately.

4.2. Multi-granular Graph Representation

Recognizing that a single document may encapsulate multiple layers of information, namely, document-level, sentence-level, and word-level, we propose the modeling of the text corpus using a multi-granular graph structure. This structure encompasses three types of vertices: document nodes, sentence nodes, and word nodes, each initialized with an embedding of the label class.

In order to fully capture the multi-granular information, we only utilize the sentence- and keyword-level representation to formulate document representation. Thus, we create a bipartite sentence-keyword graph, $\mathcal{B} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of sentence and keyword nodes, $|\mathcal{V}| = |\mathcal{S}| + |\mathcal{K}|$, and $< S_i, K_j > \in \mathcal{E}$ indicates an undirected edge between the sentence node S_i and the keyword node K_j.

4.3. Contrastive Learning on Graph

In this module, our focus in on leveraging the inherent structural information of multi-granular graphs via contrastive learning, to enhance the representative power of sentence and keyword nodes prior to pseudo-label guided training. Motivated by the findings of [7] and [10] that two connected nodes are more likely to share fine-grained topics than two randomly picked paragraphs, our goal is to pre-train the embedding space to ensure that connected sentence and keyword nodes, which are more probable to have similar class labels, are positioned closer in the embedding space, compared to a pair of randomly selected nodes.

To implement this, we have formulated a contrastive learning method based on the InfoNCE loss function.

Specifically, this method involves training samples, comprising an anchor sentence node, a positively connected keyword node, and a set of negatively unconnected keyword nodes, denoted as s, k^+, and $K^- = \{k_i^-\}_{i=1}^N$, respectively. The objective of the contrastive learning is to predict the cosine similarity between s and k^+, as well as between s and each k^-, thereby effectively distinguishing the positive sample from the negative samples.

The InfoNCE loss ($\mathcal{L}_{\text{NCE}}$) is calculated as the negative expectation of the logarithm of the similarity score ratio.

$$\mathcal{L}_{\text{NCE}} = -\mathbb{E}\Big[\log \frac{\text{score}(f(s), f(k^+))}{Z}\Big]$$

$$Z = \text{score}(f(s), f(k^+)) + \sum_i \text{score}(f(s), f(k_i^-))$$

Here, score(x, y) measures the cosine similarity between x and y, and $f(\cdot)$ denotes the function for generating node representations. Z is the partition function, which is the sum of the similarity score of the anchor and positive pair plus the sum of the similarity scores of the anchor and each negative sample.

By optimizing the InfoNCE loss, our proposed method facilitates to enhance the embedding space such that it becomes more sensitive to the indicative connections of shared class labels, thereby improving the representative power of the sentence and keyword nodes within the multi-granular graph.

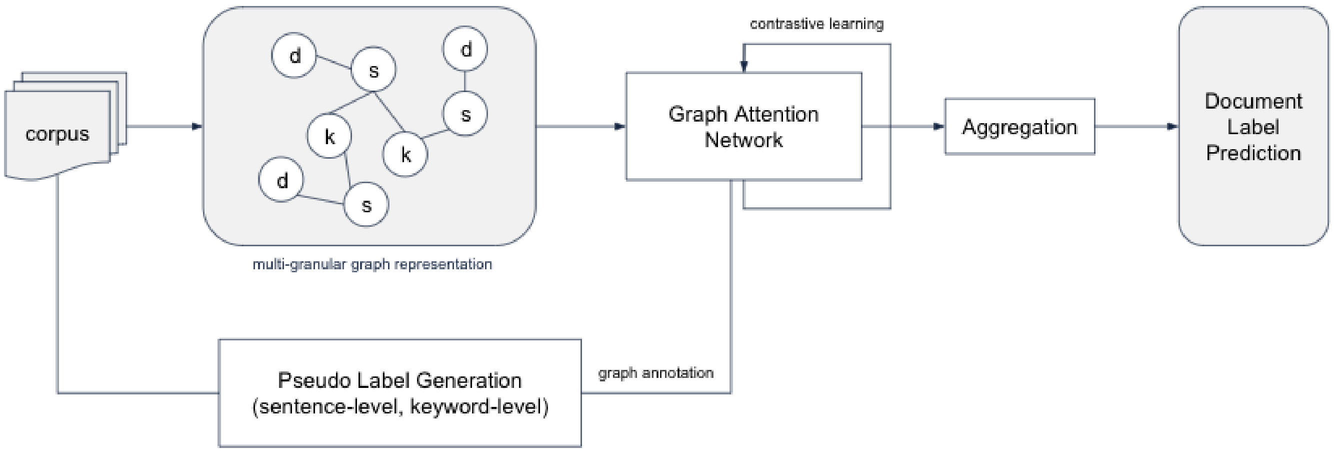

corpus
d
s
d
s
k
k
d
s
multi-granular graph representation
contrastive learning
Graph Attention Network
Aggregation
Document Label Prediction
Pseudo Label Generation (sentence-level, keyword-level)
graph annotation

Figure 2: Overview of the proposed framework. This illustrates the workflow from raw corpus input to final document label prediction. The framework begins with the corpus being subjected to pseudo-label generation, where sentence- and keyword-level pseudo-labels are created. These labels then annotate the multi-granular graph representation and the graph is processed through a Graph Attention Network, which applies contrastive learning and pseudo-label guided training to enhance the node representation, ultimately used for document label prediction by aggregating sentence-level representation.

Dataset	# Train	# Test	# Classes	# Sents	# KeyWs
Amazon-531	29,487	19,685	531	160,953	3,030

TABLE 1: Dataset statistics. Contrastive learning is conducted on the entire training set. Weakly-supervised methods are trained on training set with pseudo-labels only. Self-training methods are conducted on unlabeled training set.

4.4. Pseudo-label Guided Classifier Training

In this section, we delve into the network architecture and training methodology of two classifiers for hierarchical multi-label text classification. These classifiers are trained using the pseudo-labels generated in the preliminary phase of our framework.

4.4.1. Network Architecture. Our network architecture is designed to accommodate the complexities of hierarchical multi-label classification.

One of the key challenges addressed by our architecture is capturing the relative importance of keywords in the context of different sentences (vice versa). For example, consider the keyword ”apple” in two different sentences: ”The apple is ripe and ready to eat” and ”The latest Apple device launches today.” Although the keyword is identical, its significance and meaning differ drastically between the two contexts, influencing the classification based on the connected sentences.

To address this challenge, we incorporate attention mechanisms within our network, specifically by using a Graph Attention Network (GAT) [6]. The GAT calculates the attention coefficients that effectively determine the importance of nodes relative to each other, thereby transforming input features into higher-level features that capture the essence of their connections and distinctions. The attention coefficients e_{ij} are computed by the GAT's self-attention mechanism a, which processes the features h_i and h_j of two nodes through a shared weight matrix $\mathbf{W}$. This mechanism assigns importance to node features within the graph, taking into account their mutual influence.

$$e_{ij} = a(\mathbf{W}h_i, \mathbf{W}h_j)$$

To calculate the attention score α_{ij} for each pair of nodes, the GAT applies the softmax function to normalize exponential scores, which are derived from the LeakyReLU activation of the concatenated and transformed features of the nodes. This attention score determines the degree to which node i influences node j.

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^{\top}[\mathbf{W}h_i || \mathbf{W}h_j]))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(\mathbf{a}^{\top}[\mathbf{W}h_i || \mathbf{W}h_k]))}$$

However, different from the standard GAT structure, our model introduces two separate prediction layers for sentence and keyword nodes. We first designate indices for sentence and keyword nodes, channeling sentence nodes to a multi-label classifier and keyword nodes to a multi-class classifier. The outcomes are then indexed back to their original positions in the output layer. This dual-classification approach enables our network to discern and appropriately weigh the distinct roles that sentences and keywords play in the classification process, resulting in a more nuanced and context-aware classification outcome.

4.4.2. Training Method. The training methodology of our classifiers incorporate a strategic approach to dealing with the unique combinations of multi-class and multi-label classification tasks associated with multi-granular bipartite graphs.

1. **Multi-Class Classification of Keyword Nodes**: We utilize cross-entropy loss, a standard choice for multi-class classification problems, to train the classifier for keyword nodes. This loss functions measure the performance of the classification model whose output is a probability value between 0 and 1. The goal is to minimize the difference between the predicted probability and the pseudo-label, with the assumption of high-quality on pseudo-labels.

$$\mathcal{L}_{\text{keyword}} = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

Here, M is the number of classes, y is a binary indicator (0 or 1) if class label c is the correct classification for observation o, and p is the predicted probability observation o is of class c.

2. **Multi-Label Classification of Sentence Nodes**: For sentence nodes, which require multi-label classification, we employ binary cross-entropy loss. This is an appropriate loss function for multi-label problems, where each label is treated independently, allowing for the classification of multiple non-exclusive classes.

$$\mathcal{L}_{\text{sentence}} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \cdot \log(\hat{y}_i) + (1-y_i) \cdot \log(1-\hat{y}_i)]$$

In this equation, N is the number of observations, y_i is the true label for observation i, and $\hat{y}_i$ is the predicted probability that observation i is positive for the given label.

3. **Loss Combination**: We compute two distinct types of losses - sentence loss and keyword loss. These are then combined into a single loss function with predefined weights, balancing the contribution of each classifier to the overall training objective.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{sentence}} + \mathcal{L}_{\text{keyword}}$$

4. **Handling Imbalance with Positive Weights**: To effectively manage the imbalance in training dataset, we apply positive weights to the binary cross-entropy loss for sentence nodes. This adjustment increases the influence of less frequent, yet significant classes in the loss calculation, promoting a more balanced learning process.

5. **Self-training Phase**: A self-training phase is integrated into the training phase. At specified intervals, we assess the performance of the model on the dataset that lacks pseudo-labels. This evaluation includes the unlabeled data based on the confidence of the model's predictions. For unlabeled keyword nodes, when the highest predicted probability for a keyword node exceeds the threshold for multi-class classification, we consider this as a high-confidence prediction and incorporate the corresponding data into the training set. For unlabeled sentence nodes, sentence nodes with a number of predicted classes exceeding the threshold - typically between 1 and 3 - are also added to the training dataset. This criterion ensures that only sentence nodes with a moderate level of classification confidence are selected, avoiding the potential noise of too many predicted labels.

5. Experiment

5.1. Datasets

For our experiment, we utilize the publicly available dataset from [6]. The dataset, **Amazon-531**, includes a total of 49.145 products reviews and a three-level class taxonomy consisting of 531 classes. This dataset has been pre-processed to be all lower-cased and truncated to has maximum 500 tokens. Subsequently, we decompose each document into sentences and extract pertinent keywords from each sentence by using AutoPhrase [5] to highlight single-word phrases with phrase score greater or equal to one and multi-word phrases with phrase score greater or equal to 0.7. The statistical breakdown of the dataset is in Table 1.

5.2. Preliminary Findings

To evaluate the efficacy of pseudo labels, two critical considerations are typically addressed: 1) the representativeness of pseudo labels with respect to the text, and 2) the extent of overlap between pseudo labels and true document labels. Consequently, a comprehensive analysis is conducted, encompassing both quantitative and qualitative dimensions. Quantitatively, we aggregate the set of all pseudo labels within the document and compute the proportion of accurately identified labels. For qualitative analysis, a subset of labeled documents is randomly sampled, and the representativeness of pseudo labels manually analyzed and summarized.

5.2.1. Keyword Pseudo Labels. Regarding the model parameters used in Seetopic, an embedding size of 100 is utilized, and words occurring fewer than three times are excluded. Two pretrain iterations and four ensemble iterations are conducted to derive class-indicative keywords, resulting in 12 keywords for each label.

As illustrated in Figure 3. keyword pseudo label sets on average captures less than 1 positive for each document. In addition, the number of negative keywords exceeds that of the positive keywords. Upon scrutinizing the corpus further, our observation suggests that an excess generation of words related to the label but not intrinsic to the label may induce semantic drift.

5.2.2. Sentence Pseudo Labels. In the TaxoClass framework, the selection of confident pseudo labels for documents is contingent upon the confidence score, serving as a metric for (undetermined property). Deviating from the

Average positive label #		Metric	
		probability	confidence
Strategy	By level	1.15	0.90
	By labels	1.05	0.84

TABLE 2: Average number of positive labels captured by four filtering strategies

Average negative label #		Metric	
		probability	confidence
Strategy	By level	2.6	3.13
	By labels	3.03	3.33

TABLE 3: Average number of negative labels captured by four filtering strategies

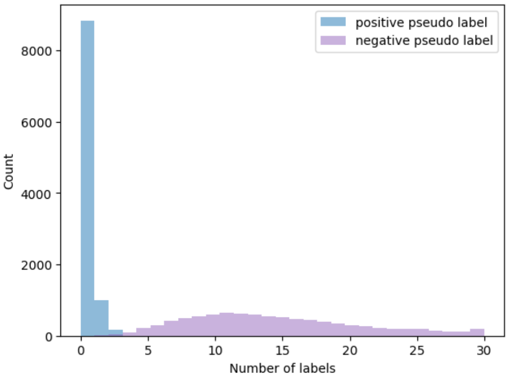


Figure 3: Distribution of positive label number and negative label number within in the keyword pseudo label set

original approach, our methodology employs an alternative textual entailment PLM model for calculating sentence-class probability. Furthermore, our focus is on generating pseudo labels for sentences rather than classes, potentially resulting in distributional disparities. Consequently, we propose four filtering strategies based on two dimensions:

- **By level vs. By label:** Under the assumption that pseudo label distribution varies across labels and hierarchy levels, we seek to ascertain whether retaining the top 10% labels within the same class or opting for the top 10% labels within the same level hierarchy is more efficacious.
- **Confidence score vs. Probability score** In the selection of confident core classes, two variables measuring class similarity—probability score are used. The sentence-label similarity score is directly generated by the textual entailment PLM and confidence score is defined as the relative sentence similarly when being compared to neighbouring classes. Our objective is to discern which metric yields more representative labels.

As illustrated in Table 3. all filtering strategies successfully capture approximately one correct document labels, with the strategy of filtering achieving the best performance where 1.15 positive labels and 2.6 negative labels are captured by the pseudo label set on average. Notably, through manually examining documents and summarizing the reasons causing pseudo label classification errors, the observed enhancement in performance is attributed to a reduction in the likelihood of incorporating completely entirely erroneous pseudo labels into the set of confident pseudo labels.

5.3. Compared Methods

Given that our proposed approach is principally founded on the methodologies established by [6], it is essential to benchmark the performance of our variant models against their framework:

- **TaxoClass** [6]: Employing a weakly-supervised framework, TaxoClass addresses the challenges posed by a large label space and the scarcity of labeled data. It utilizes a hierarchical organization of classes within a taxonomy to facilitate multi-label tagging by capturing class relations.

- **Our Method**: Our proposed weakly-supervised framework formulates a multi-granular graph representation and generates pseudo-labels for sentence nodes, conduct contrastive learning and pseudo-label guided sentence classifier training to be further utilized for document label prediction. **Baseline** only utilizes the generated pseudo-labels and attention mechanism of our network architecture. **Baseline + CL** adopts InfoNCE based contrastive learning on top of Baseline. **Baseline + SSL** adopts self-supervised learning on top of Baseline. Finally, **Baseline + CL + SSL** adopts both contrastive learning and self-supervised learning on top of the Baseline model.

5.4. Evaluation Metrics

To ensure consistency and comparability of our evaluation, we also follow the same metrics used by [6] for assessing multi-label classification performance. Our evaluation involves with a variery of metrics that examine the results with different perspectives. The first metric employed is **Example-F1**, which calculates the averge F1 scores across all documents and is computed as follows:

$$\text{Example-F1} = \frac{1}{N}\sum_{i=1}^{N}\frac{2|C_i^{\text{true}} \cap C_i^{\text{pred}}|}{|C_i^{\text{true}}| + |C_i^{\text{pred}}|}$$

In this formula, N represents the total number of documents, C_i^{true} denotes the set of true class labels for the i-th document, and C_i^{pred} represents the set of predicted class labels for the same document.

Additionally, as many applications approach Hierarchical Multi-Label Text Classification a a class ranking problem, we adapt our evaluation strategy accordingly. This involves converting the predicted class set C_i^{pred} into a ranked list R_i^{pred} based on the predicted probability of each class from the model. To evaluate the effectiveness of this ranking, we calculate the **Precision at** k metric as follows:

$$P@k = \frac{1}{N}\sum_{i=1}^{N}\frac{|C_i^{\text{true}} \cap R_{i,1:k}^{\text{pred}}|}{\min(k, |C_i^{\text{true}}|)}$$

In this equation, $R_{i,1:k}^{\text{pred}}$ refers to the top k most likely classes predicted for document D_i, by each method. P@k thus measures the precision of the model in predicting the top k classes, considering the actual true classes for each document. This metric offers a focused view of the model's accuracy in identifying the most relevant classes for each document.

Lastly, we compute **Mean Reciprocal Rank** (MRR). The MRR offers insights into how well the model ranks the true classes of a document, particularly in situations where multiple classes are equally valid. The formula for calculating the MRR is as follows:

$$\text{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|C_i^{\text{true}}|}\sum_{c_j \in C_i^{\text{true}}}\frac{1}{R_{ij}}$$

In this equation, R_{ij} represents the rank of the true class c_j of document D_i within the model's predicted ranking list across all classes. By averaging the reciprocal ranks of the true classes, the MRR provides a measure of the model's ability to not only correctly classify documents but also to prioritize the most relevant classes higher in its ranking.

5.5. Experiment Settings

For training our model, we use Adam optimizer. We employ a learning rate of 0.01 for all parameters involved in the sentence classifier training, while a lower rate of 1e-4 is applied for the contrastive learning process. The initial setup for Graph Attention Network (GAT) incorporates a preliminary layer featuring a hidden layer dimension of 128, 8 attention heads, and a dropout rate of 0.3 to prevent overfitting.

We execute the self-supervised learning phase at intervals of every 25 epochs. In the context of contrastive learning's computation of the InfoNCE loss, we incorporate 10 randomly selected negative samples for each anchor node to enhance the learning efficacy. The output dimension for the contrastive learning is set at 768, aligning it with the dimensions of sentence and keyword embeddings.

For the generation of sentence, we utilized ‘all-mpnet-base-v2‘ model, while ‘bert-base-uncased‘ model is employed for keyword embeddings. These choices are made to ensure a consistent embedding dimension of 768 across both types of embeddings.

We conduct all our experiments on an A100 GPU via Google Colab. The contrastive learning framework demonstrates a substantial memory footprint, consuming approximately 15GB of GPU memory. The pseudo-label guided sentence classifier training exhibited a more modest average memory usage of around 10GB.

5.6. Overall Performance Comparison

Table 2 presents the comparative results of all the methods we experimented with. We observe that the overall performance of our proposed method did not meet expectations, especially against TaxoClass. While the incorporation of the contrastive learning framework into the baseline model did result in a marginal improvement in performance, the introduction of self-supervised learning, contrary to our expectations, leads to a decrease in performance relative to the baseline model. In the following sections, we delve into a detailed examination of the underlying factors contributing to the under-performance of our proposed model, exploring potential explanations and considering avenues for refinement and improvement of our approach.

5.6.1. Low-quality of keyword's pseudo-labels. Through a rigorous empirical examination of keyword pseudo labels, it becomes evident that the primary factor contributing to the suboptimal quality of these labels lies in the propensity of SeeTopic to generate words that are semantically similar to label names but don't belong to the label class, ultimately resulting in content drifting. We hypothesize current pseudo labelling procedure is not efficiently utilizing the hierarchical information in label spaces, resulting in the label spaces being too large and not well-separated.

Consequently, potential strategies for enhancing the quality of keyword pseudo labels includes the development of a hierarchical topic mining algorithm akin to JoSH [4], but is robust under out-of-vocabulary label surface names. Alternatively, the incorporation of category-indicative metrics, as proposed in the LOTClass [3], offers a promising approach to mitigate the presence of low-quality keywords.

5.6.2. Extremely Imbalanced Sentence Dataset. Through the investigation into the under-performance of our proposed model, we conducted thorough analysis focusing on the distribution of pseudo-labels across the classes in the Amazon-531 dataset, in comparison to the actual distribution of true labels. This led to several notable discoveries. First, we found a significant disparity of the pseudo-labeled sentence dataset when compared the true document dataset. The sentence dataset exhibited a much higher level of sparsity. Specifically, a 11.5% of the classes in the pseudo-label sentence dataset did not have any positive samples, contrast to just 3.5% in true label document dataset.

Method	Example-F1	P@1	P@3	MRR
TaxoClass	0.5934	0.8120	0.5894	0.6332
Baseline	0.0161	0.0208	0.0161	0.0255
Baseline + CL	**0.0228**	**0.0251**	**0.0231**	**0.0305**
Baseline + SSL	0.0118	0.0097	0.0119	0.0217
Baseline + CL + SSL	0.0139	0.0147	0.0139	0.0225

TABLE 4: Evaluation of our proposed method and TaxoClass on Amazon-531.

Second, in terms of positive sample distribution across class labels, the pseudo-label sentence dataset lagged considerably behind the true label dataset. Only a 3.95% of the class labels in the pseudo-label sentence dataset accounted for more than 1% of positive samples, compared to 11% in the true label document dataset.

Third, a further observation was none of the pseudo-labels in sentence dataset accounted for more than 10% of positive samples. This absence of high-positive labels indicates an imbalance in the dataset.

These findings highlight a critical challenge in our approach: the extreme imbalance and sparsity in the pseudo-labeled sentence dataset. This imbalance likely contributed to the model's inability to effectively learn and generalize across the diverse range of classes, ultimately impact its overall performance.

Addressing this imbalance is crucial for enhancing the model's ability to learn effectively from the pseudo-labels and improve overall performance. Two promising approaches are under consideration. First, introducing L1 regularization to our loss function could be an effective strategy. L1 regularization, known for its feature selection capabilities, could help in reducing the impact of less informative features while emphasizing more significant ones. This approach might contribute to a more balanced learning process and improved model generalization. Second, inspired by [12] on graph oversampling, we consider the adaptation of similar techniques in our framework. GraphSMOTE is a novel method specifically dealing with imbalances in graph-structured data. With the consideration of intensive relationship within the graph, we could create a more balanced dataset, thereby providing a more equitable training ground for the model. Our future work will involve a detailed exploration and implementation of these methods, followed by testing to assess their impact on model performance.

6. Conclusion

This research explored into the domain of Weakly-supervised Hierarchical Multi-label Text Classification (HTMC) with a focus on leveraging multi-granular text units, encompassing both keywords and sentences. Our approach established a pseudo-label generation process tailored to these text units, followed by a multi-granular graph representation enriched with attention mechanisms. This representation is designed to capture the complex relationships and hierarchical structures inherent in text data.

Despite the innovative approach the theoretical promise of our proposed methodology, the performance of the model is found to be sub-optimal. This shortfall was primarily rooted from the challenges posed by low-quality pseudo-labels and the imbalances present in the sentence-level data. These factors collectively bordered the model's ability to accurately classify and predict labels in a hierarchical, multi-label context.

In light of these challenges, our future research will focus on (1) enhancing pseudo-label quality by introducing ”category indicative” measures to reduce the amount of low-quality keywords, and (2) handling the imbalance on dataset by the implementation of L1-regularization on the loss and the adaption of graph oversampling techniques. Through these focused efforts, we anticipate advancements in the field of weakly-supervised HMTC, with the goal of developing a more robust, accurate, and reliable classification model capable of handling the complexities of large-scale multi-label text data.

References

[1] Priyanka Kargupta, Tanay Komarlu, Susik Yoon, Xuan Wang, and Jiawei Han. Megclass: Text classification with extremely weak supervision via mutually-enhancing text granularities. *arXiv preprint arXiv:2304.01969*, 2023.

[2] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. Weakly-supervised hierarchical text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6826–6833, 2019.

[3] Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. Text classification using label names only: A language model self-training approach. *arXiv preprint arXiv:2010.07245*, 2020.

[4] Yu Meng, Yunyi Zhang, Jiaxin Huang, Yu Zhang, Chao Zhang, and Jiawei Han. Hierarchical topic mining via joint spherical tree and text embedding. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1908–1917, 2020.

[5] Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1825–1837, 2018.

[6] Jiaming Shen, Wenda Qiu, Yu Meng, Jingbo Shang, Xiang Ren, and Jiawei Han. Taxoclass: Hierarchical multi-label text classification using only class names. In *NAAC'21: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,*{*NAACL-HLT*} *2021*, volume 2021, 2021.

[7] Michihiro Yasunaga, Jure Leskovec, and Percy Liang. Linkbert: Pretraining language models with document links. *arXiv preprint arXiv:2203.15827*, 2022.

[8] Lu Zhang, Jiandong Ding, Yi Xu, Yingyao Liu, and Shuigeng Zhou. Weakly-supervised text classification based on keyword graph. *arXiv preprint arXiv:2110.02591*, 2021.

[9] Yu Zhang, Bowen Jin, Xiusi Chen, Yanzhen Shen, Yunyi Zhang, Yu Meng, and Jiawei Han. Weakly supervised multi-label classification of full-text scientific papers. *arXiv preprint arXiv:2306.14003*, 2023.

[10] Yu Zhang, Bowen Jin, Xiusi Chen, Yanzhen Shen, Yunyi Zhang, Yu Meng, and Jiawei Han. Weakly supervised multi-label classification of full-text scientific papers. *arXiv preprint arXiv:2306.14003*, 2023.

[11] Yu Zhang, Yu Meng, Xuan Wang, Sheng Wang, and Jiawei Han. Seed-guided topic discovery with out-of-vocabulary seeds. *arXiv preprint arXiv:2205.01845*, 2022.

[12] Tianxiang Zhao, Xiang Zhang, and Suhang Wang. Graphsmote: Imbalanced node classification on graphs with graph neural networks. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 833–841, 2021.