Learning Label Hierarchy with Supervised Contrastive Learning

Ruixue Lian **William A. Sethares** **Junjie Hu**
University of Wisconsin-Madison
{ruixue.lian, sethares, junjie.hu}@wisc.edu

Abstract

Supervised contrastive learning (SCL) frameworks treat each class as independent and thus consider all classes to be equally important. This neglects the common scenario in which label hierarchy exists, where fine-grained classes under the same category show more similarity than very different ones. This paper introduces a family of Label-Aware SCL methods (LASCL) that incorporates hierarchical information to SCL by leveraging similarities between classes, resulting in creating a more well-structured and discriminative feature space. This is achieved by first adjusting the distance between instances based on measures of the proximity of their classes with the scaled instance-instance-wise contrastive. An additional instance-center-wise contrastive is introduced to move within-class examples closer to their centers, which are represented by a set of learnable label parameters. The learned label parameters can be directly used as a nearest neighbor classifier without further finetuning. In this way, a better feature representation is generated with improvements of intra-cluster compactness and inter-cluster separation. Experiments on three datasets show that the proposed LASCL works well on text classification of distinguishing a single label among multi-labels, outperforming the baseline supervised approaches. Our code is publicly available.[1]

1 Introduction

Supervised contrastive learning (SCL) (Khosla et al., 2020) aims to learn generalized and discriminative feature representations given labeled data. It relies on the construction of positive pairs from the same class and negative pairs from different classes, thereby encouraging similar data points to have similar representations while pushing dissimilar data points apart in the feature space. This method considers each class to be independent and considers all classes to be of equal importance, thus treating the problem without awareness of any relationships among the labels. However, in the real world, it is natural that class labels may relate to each other in complex ways, in particular, they may exist in a hierarchical or tree structure (Małkiński and Mańdziuk, 2022; Demszky et al., 2020; Murdock et al., 2016; Verma et al., 2012; Han et al., 2018). Within a data hierarchy, different sub-categories under the same branch tend to be more similar than those from different branches, since they will tend to have similar high-level semantics, sentiment, and structure. This similarity should be reflected in the feature representations.

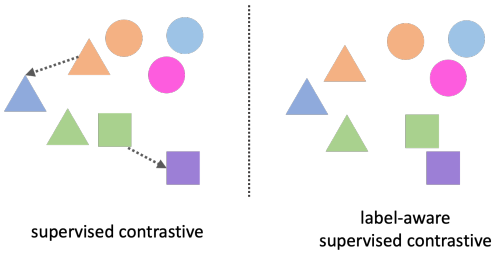


Figure 1: Supervised v.s. label-aware supervised contrastive loss: The supervised contrastive loss (left) contrasts the set of all samples from the same class as positives against the negatives from the remainder of the batch (Khosla et al., 2020). The label-aware supervised contrastive loss (right) proposed in our work incorporates label hierarchy by considering class similarities.

Hierarchical text classification (HTC) is one way to structure textual data into a tree-like category or label hierarchy, representing a taxonomy of classes (Kowsari et al., 2017). Existing HTC can be divided into global and local approaches. Global approaches treat the problem as a flat classification, while local approaches build classifiers for labels at each level of the hierarchy. An et al. (2022) propose FCDC, which aims to transfer information from coarse-grained levels to fine-grained categories and thus adapt models to categories of different gran-

[1] https://github.com/rxlian/LA-SCL

Findings of the Association for Computational Linguistics: EACL 2024, pages 1569–1581

ularity. Besides, Wang et al. (2022) incorporate label hierarchy information extracted from a separate encoder. Some other works leverage additional hierarchical information (Lin et al., 2023; Long and Webber, 2022; Suresh and Ong, 2021).

Other than that, Zeng et al. (2023) augment the classification loss by the Cophenetic Correlation Coefficient (CPCC) (Sokal and Rohlf, 1962) as a standalone regularizer to maximize the correlation between the label tree structure and class-conditioned representations. Li et al. (2021) propose a ProtoNCE loss, a generalized version of the InfoNCE loss (Oord et al., 2018) to learn a representation space by encouraging each instance to become closer to an assigned prototype such as the clustering centroid. In this way, the underlying semantic structure of the data can be encoded.

Based on these studies, the hierarchical structure of the labels suggests that learning methods could be enhanced if the learning mechanism can be made aware of the class taxonomy. We explore several ways of exploiting such hierarchical relationships between classes by proposing to augment the SCL loss function as depicted in Fig. 1. Since this incorporates class taxonomy information, we call it label-aware SCL (LASCL). This is achieved by first using pairwise class similarities to scale the temperature in the SCL to encourage samples under the same branches to cluster more closely while driving apart samples with different labels under different coarse clusters. In addition, we add instance-center-wise contrastive with learned label representations as the center of the sentence embeddings from the corresponding class. These result in making sub-classes under the same coarse-grained classes closer to each other and generating more discriminative representations by making intra-class samples closer to their centers.

To utilize intrinsic information from label and data hierarchies, we encode the textual label information to be class centers and compute pairwise class Cosine similarities on top of that. This quantifies the proximity between classes and forms the basis for instantiating variations of LASCL objectives. Since the dimension of these label representations is the same as the linear classifier, we show that it can be applied directly to downstream classification without further finetuning. To the best of our knowledge, we are the first to work on leveraging the textual hierarchical label and integrating it into the SCL to improve the representations. Our methods can be transferred to various backbone models, and are simple yet effective across different datasets. The only changes we make are in the cost function so that the method can be applied in any situation where labels in a hierarchy exist.

Our contributions are summarized as follows:

- LASCL integrates label hierarchy information into SCL by leveraging the textual descriptions of the label taxonomy.
- Our method learns a structured feature space by making fine-grained categories under the same coarse-grained categories closer to each other.
- Our method also encourages more discriminative representations by improving intra-cluster compactness and inter-cluster separation.
- The learned label parameters from our method can be used directly as a nearest neighbor classifier without further finetuning.

2 Background

Problem Setup For a supervised classification task, a labeled dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ consists of N examples from a joint distribution $P_{\mathcal{XY}}$, where $\mathcal{X}$ is the input space of all text sentences, $\mathcal{Y} = \{1, ..., C\}$ is the label space, and C is the number of classes. The goal of representation learning is to use $\mathcal{D}$ to learn a feature encoder $f_\theta : \mathcal{X} \rightarrow \mathcal{Z}$ that encodes a text sentence to a semantic sentence embedding in a feature space $\mathcal{Z}$. This allows us to measure the pairwise similarity between two text sentences x_i, x_j by a similarity function $\text{sim}(x_i, x_j)$, which first projects x_i and x_j to $\mathcal{Z}$, i.e., $\mathbf{z}_i = f_\theta(x_i)$, and computes a distance between two sentence embeddings in $\mathcal{Z}$. Moreover, learning meaningful embeddings facilitates the learning of a classifier $g_\phi : \mathcal{Z} \rightarrow \mathcal{Y}$ that maps learned embeddings to their corresponding labels.

Supervised Contrastive Learning (SCL) A major thread of representation learning focuses on supervised contrastive learning (Khosla et al., 2020) that encourages embedding proximity among examples in the same class while simultaneously pushing away embeddings from different classes using the loss function in Eq. (1). Specifically, for a given example (x_i, y_i), we denote $\mathcal{P}(y_i) = \{x_j | y_j = y_i, (x_j, y_j) \in \mathcal{D}\}$ as the set of sentences in $\mathcal{D}$ having the same label as y_i. Thus, the SCL loss is computed on $\mathcal{D}$ as:

$$\ell_{\text{SCL}}(x_i, y_i) = \mathbb{E}_{x_j \sim \mathcal{P}(y_i)} \log \frac{\exp(\frac{\text{sim}(x_i, x_j)}{\tau})}{\sum_{k \notin \mathcal{P}(y_i)} \exp(\frac{\text{sim}(x_i, x_k)}{\tau})}$$

$$\mathcal{L}_\tau(\mathcal{D}; \theta) = \mathbb{E}_{(x_i, y_i) \sim \mathcal{D}}\, \ell_\tau(x_i, y_i), \quad (1)$$

The fixed hyper-parameter τ is the temperature that adjusts the embedding similarity of sentence pairs.

3 Method

This section describes our proposed label-aware supervised contrastive learning objectives.

Overview: In the embedding space, we hypothesize that sentences from different fine-grained classes under the same coarse-grained class are closer to each other in comparison to sentences from different high-level categories. Given this intrinsic information provided by the label and data hierarchy, we use the pairwise cosine similarities of a set of learnable parameters representing label features to quantify the proximity between classes, which are used to instantiate variants of label-aware supervised contrastive learning objectives.

3.1 Label Hierarchy and Class Similarities

This section describes the construction of learnable label representations given label hierarchies, which are used to calculate similarities between classes.

A label hierarchy of a labeled dataset refers to a hierarchical tree that defines an up-down, coarse-to-fine-grained structure with labels being assigned to a corresponding branch. We use label textual descriptions to construct the tree structure. Let $\mathcal{T}$ be a hierarchical tree with V being the set of intermediate and leaf nodes. Each leaf node v_c represents a class label $c \in \mathcal{Y}$, and is associated with a set of examples in class c, i.e., $\mathcal{P}(c)$, where $\mathcal{P}(c) \cap \mathcal{P}(c') = 0$, $\forall c \neq c'$. Each parent node represents a coarse-grained category containing a set of fine-grained children nodes. The leaf nodes can have different depths in $\mathcal{T}$, which refers to the distance between each leaf node v_c and root node v_0. Let L_i be the i-th layer of $\mathcal{T}$. Figure 2a shows an example of a tree-structured label hierarchy built from 20News dataset (Lang, 1995).

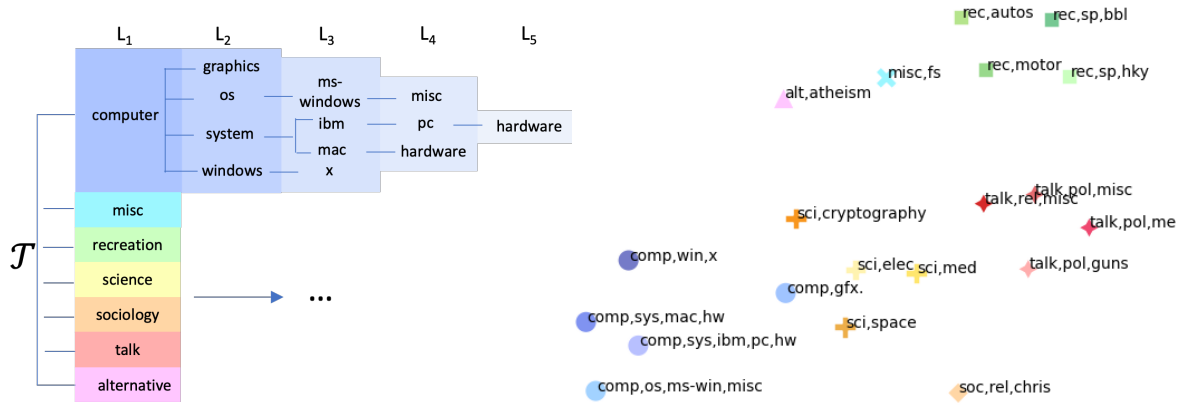
Given $\mathcal{T}$, we exploit the hierarchical relationships among the classes by having more informative descriptions. To achieve this, given a leaf node of class $c \in \mathcal{Y}$, its ancestor nodes are first collected until reaching the leaf node. These up-down textual classes at different levels are concatenated into a text sequence, which is then filled in by a sentence template. For Figure 2a, for a leaf node of "Hardware" at L_5, we collect its ancestors and assign "Computer, System, IBM, PC, Hardware" as its label. In this way, the hierarchical information of labels is collected and can be extracted by an encoder. Let u_c be a sentence of class $c \in \mathcal{Y}$. A pretrained language encoder f_θ is used to obtain a label representation denoted as $\mathbf{u}_c = f_\theta(u_c)$. This set of label representations are made of learnable parameters and will be updated during back-propagation. To stabilize the process, we re-encode the label representations less frequently than the updates of the sentence embeddings, that is, extract label embeddings only after every n iterations.

After encoding label representations for all classes $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_C]$, a pairwise cosine similarity measurement is applied to compute a class similarity matrix $\mathbf{W} \in \mathbb{R}^{C \times C}$, where each entry is the similarity score between a label c and another label c', i.e., $w_{cc'} = \text{sim}(u_c, u_{c'})$. $\mathbf{W}$ will be further applied to scale the temperature in §3.2. Note that this label embedding matrix $\mathbf{U} \in \mathbb{R}^{d \times C}$ can be directly used as a nearest-neighbor classifier, where it can be applied to linearly map an input sentence embedding $x_i \in \mathbb{R}^d$ into the label space $\mathcal{Y}$. Therefore, $\mathbf{U}$ can be applied as a linear head for the downstream classification without further finetuning.

Figure 2b shows the t-SNE (Van der Maaten and Hinton, 2008) visualization of 20 initialized label embeddings of the 20News extracted from their sentence description encoded by a pretrained BERT-base model. Different high-level and lower-level classes are displayed with different markers and colors. Observe that labels from the same coarse-grained classes are clustered closer to each other than to other classes. Given the clustering nature of the labels reflects their hierarchical structure, these class similarities can be utilized as additional information to scale the importance of different classes, which is introduced in the next section.

3.2 Scaling with Class Similarities

This section describes a way to incorporate the class hierarchy information into supervised contrastive loss by leveraging additional scalings introduced in $\mathbf{W}$. The overall idea is to scale the temperature τ in Eq. (1) by $\mathbf{W}$, which reflects similarities between classes and is updated every several iterations. Specifically, the negative example pairs in



(a) (b)

Figure 2: (a) The label hierarchy of the 20News dataset. The root node contains 7 classes, each branch has multiple fine-grained sub-categories. (b) t-SNE visualization of hierarchical label embeddings encoded by BERT-base.

SCL are weighted by the corresponding learned class similarities, performing a scaled instance-to-instance update. The final loss over a dataset $\mathcal{D}$ is the same form as Eq. (1) with the individual loss ℓ_τ replaced by

$$\ell_{sii}(x_i, y_i) = \mathbb{E}_{j\sim\mathcal{P}(y_i)} \log \frac{\exp\left(\frac{\text{sim}(x_i,x_j)}{\tau}\right)}{\sum_{k\notin\mathcal{P}(y_i)} \exp\left(\frac{\text{sim}(x_i,x_k)}{\tau\cdot s_{ik}}\right)}, \quad (2)$$

where the elements of the matrix $\mathbf{W}$ define the pairwise similarity between labels, abbreviated by $s_{ik} = w_{y_i,y_k}$ for a label pair y_i and y_k.

In this way, Eq. (2) scales the similarity between negative pairs based on the similarity between the corresponding classes. Consider two samples x_i and x_k from different classes y_i and y_k. The similarity s_{ik} tends to be greater if y_i and y_k have the same parent category. Thus, it applies a higher penalty to the negative pairs when they are from different coarse-grained categories, so the learning update tends to push them further apart. In this way, the label hierarchical information is introduced to assign different penalties, reflecting the similarities and dissimilarities between classes.

3.3 Label Representations as Class Centers

The label representations can also be used as class centers to perform instance-center-wise contrastive learning, as shown in another loss term ℓ_{ic}.

$$\ell_{ic}(x_i, y_i) = \log \frac{\exp\left(\frac{\text{sim}(x_i,u_{y_i})}{\tau}\right)}{\sum_{k\notin\mathcal{P}(i)} \exp\left(\frac{\text{sim}(x_i,u_{y_k})}{\tau}\right)}. \quad (3)$$

This loss term ℓ_{ic} regards the label sequence u_c constructed for the label c as the center of the sentences from this class. Thus, for each input instance x_i, a positive pair is constructed between the instance and its center as (x_i, u_{y_i}), and negative pairs are constructed by comparing the instance x_i with other label sequences, $(x_i, u_{y_k}), \forall y_k \neq y_i$. This loss function pulls each sentence closer to its label center and further from other centers, thus making each cluster more compact in the embedding space.

Similarly to Eq. (2), the temperature in ℓ_{ic} can be scaled by the class similarity s_{ik}, and thus we can construct a scaled instance-center-wise contrastive loss term as follow:

$$\ell_{sic}(x_i, y_i) = \log \frac{\exp\left(\frac{\text{sim}(x_i,u_i)}{\tau}\right)}{\sum_{k\notin P(i)} \exp\left(\frac{\text{sim}(x_i,u_k)}{\tau\cdot s_{ik}}\right)}. \quad (4)$$

3.4 Label-Aware SCL Variants

Based on the aforementioned loss functions, we propose four label-aware SCL (LASCL) variants and compare their performance in §5.

Label-aware Instance-to-Instance (LI) The first variant is shown in Eq. (2), which modifies the original SCL by scaling the temperature by the label similarity.

Label-aware Instance-to-Unweighted-Center (LIUC) The second variant augments the original SCL by adding an unweighted instance-center-wise contrastive loss.

$$\ell_{\text{LIUC}} = \ell_{\text{SCL}} + \ell_{ic} \quad (5)$$

Label-aware Instance-to-Center (LIC) The third variant augments our first variant by adding an unweighted instance-center-wise contrastive loss.

$$\ell_{\text{LIC}} = \ell_{sii} + \ell_{ic} \quad (6)$$

Label-aware Instance-to-Scaled-Center (LISC) The final one augments our first variant by adding a weighted instance-center-wise contrastive loss.

$$\ell_{\text{LISC}} = \ell_{sii} + \ell_{sic} \tag{7}$$

4 Experimental Settings

Dataset	train/val/test (original) (K)	train/val/test (LP) (K)	classes ($\|L_1\|/\|L_n\|$)
20News	10/1/7	2/2/7	7/20
WOS	38/4/4	1/1/4	7/134
DBPedia	238/2/60	12/12/60	9/70

Table 1: Dataset statistics. $|L_1|$ and $|L_n|$ are number of coarse-grained and fine-grained classes, respectively.

Datasests 20NewsGroups[2] (news classification) (Lang, 1995), WOS (paper classification) (Kowsari et al., 2017), DBPedia (topic classification)(Auer et al., 2007), and their originally provided label structures and textual labels are used in our experiments. Each leaf node label of 20News has different depth, while each leaf node lable of WOS and DBPedia have the same depth 2. Dataset statistics is shown in Table 1. For linear-probe (LP) experiments, we randomly select samples with balanced distribution.

Sentence Templates We use the following templates to fill in the label string for each dataset, which is further encoded by a BERT model.

- 20News: "It contains {label$_i$} news."
- WOS: "It contains article in domain of {label$_i$}."
- DBPedia: "It contains {label$_i[L_2]$} under {label$_i[L_1]$} category."

Implementation Details We use *bert-base-uncased* provided in huggingface's packages (Wolf et al., 2019) as our backbone models. The averaged word embeddings of the last layer are used as sentence representations. We used learning rate 1e-5 with linear scheduler and weight decay 0.1. The model is trained with 20 epochs and validated every 256 steps. To avoid overfitting, the best checkpoints were selected with an early stop and patience of 5 according to evaluation metrics. For LP, we use a learning rate of 5e-3 with a weight decay of 0.01. The classifier was trained with 10 epochs and validated after each epoch. The best checkpoint was selected according to validation accuracy. The batch size and max sequence length are 32 and 128, respectively, across all the experiments. The temperature τ is 0.3. During training, we re-encode the label embeddings every 500 steps. Cosine similarity was used over all experiments.

Evaluation Metrics We report: (1) classification accuracy on the leaf node called **nodeAcc** (2) classification accuracy on the parent node of the leaf, which is called **midAcc**, (3) classification accuracy on the root node, which is the highest level of each branch and is called **rootAcc**.

5 Results and Analysis

To demonstrate the effect of the amount of labeled data to LASCL, we perform experiments with both the few-shot setup and full dataset in §5.1 and §5.2. In §5.3, we visually show how the proposed methods generate a more well-structured and discriminative embedding space by visualizations. We discuss how the size of the hierarchy plays a role by constructing a bottom-up label hierarchy with different depths in §5.4.

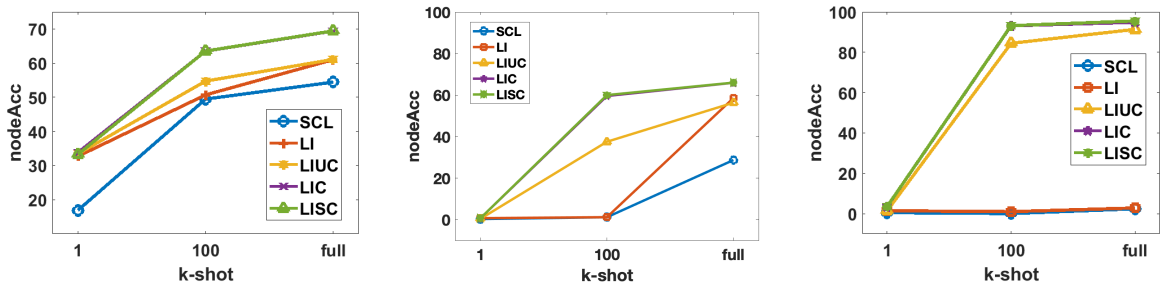
The experimental results are reported with linear probes (LP) and with direct testing (DT). For LP, a randomly initialized linear layer was trained on a small number of labeled samples with the encoder frozen. We denote DT as directly applying the learned label parameters as the classifier (§3.4).

5.1 Few-Shot Cases

LASCL works well on few-shot cases. We first conduct k-shot experiments with k=1 and k=100. To be specific, we take 1 and 100 sentences from each class to construct the training set. The validation and test sets remain the same as the original. NodeAcc on direct testing experiments are shown in Figure 3, and the accuracies are summarized in Table 6 in the Appendix.

We can observe improvements under few-shot cases by applying LASCL across three datasets, while there are some differences in terms of hierarchical label granularities reflected by the datasets. LI is effective when there exists a more comprehensive label hierarchical information as shown in Fig. 3a, where 20News has a deeper hierarchy of fine-grained labels compared to DBPedia and WOS (Fig. 3c and 3b) which have only two layers for each label. It indicates that a more comprehensive hierarchy that captures the intricate relationships between classes would be more beneficial.

[2] http://qwone.com/~jason/20Newsgroups/



(a) 20News (b) WOS (c) DBPedia

Figure 3: Directly testing (DT) the k-shot prediction performance (measured by NodeAcc) on three datasets.

Dataset	Objective	direct test			linear probe		
		nodeAcc	midAcc	rootAcc	nodeAcc	midAcc	rootAcc
20News	SCL	54.44	61.74	69.41	65.64	72.54	78.98
	LI	61.01	67.19	73.09	67.59	74.04	79.82
	LIUC	61.09	69.62	79.17	66.42	73.66	79.67
	LIC	69.40	75.64	81.05	68.32	75.21	80.87
	LISC	**69.45**	**75.90**	**81.08**	**68.47**	**75.33**	**81.07**
WOS	SCL	28.71	–	46.50	54.03	–	70.06
	LI	58.57	–	70.91	62.14	–	74.97
	LIUC	56.35	–	71.89	58.32	–	72.89
	LIC	65.97	–	78.46	73.17	–	83.12
	LISC	**66.02**	–	**78.47**	**73.56**	–	**83.13**
DBPedia	SCL	2.42	–	38.26	96.00	–	96.79
	LI	2.84	–	31.25	96.14	–	96.80
	LIUC	91.34	–	94.65	96.00	–	96.79
	LIC	94.85	–	96.30	96.52	–	97.25
	LISC	**95.52**	–	**97.06**	**96.71**	–	**97.35**

Table 2: Classification accuracy (%) in terms of the leaf, mid-layer, and root nodes with models trained on SCL, LI, LIUC, LIC, and LISC on 20News, WOS, and DBPedia datasets.

Besides, LIC, LIUC, and LISC, which incorporate additional contrastive objectives between instances and centers, achieve notable performance and largely close the gap, especially between full dataset and 100-shot on DBPedia and WOS datasets. It effectively utilizes the label information even if the hierarchical structure is shallow. With 100-shot, the computation cost is decreased by reducing the training set size to 1% while maintaining decent performance compared to with full dataset.

5.2 Full Dataset

LASCL outperforms SCL in full-data setting. Table 2 shows the results on the full dataset with our proposed four LASCL objectives, which outperform SCL in terms of the accuracy on the leaf node, mid-layer, and root level metrics for both DT and LP experiments. In most cases, LP enhances the performance compared to DT, while maintaining a comparable performance across different objectives. The performance gain introduced by LIC and LISC is substantial enough to narrow the performance gap between DT and LP. In particular, DT performs better than LP on 20News, indicating the creation of effective label representations.

Among the four proposed variants, the additional scaling introduced by the class similarities contribute to the performance gains, especially when dealing with fine-grained hierarchies. The improvement is clearest using the nodeAcc test comparing SCL and LI where the accuracy is increased by effectively penalizing the distance be-

(a) bert-base (b) SCL (c) LISC

Figure 4: t-SNE visualization on 20News dataset (keep the original distribution) with (a) bert-base, (b) SCL, (c) LISC. Label representations are marked by appropriately colored "×".

tween classes. Moreover, compared to SCL, the additional instance-center-wise contrastive loss introduced by LIUC also induces performance gains, especially on rootAcc of coarse-grained categories. It leads to clearer decision boundaries between coarse-grained categories, and moves within-class instances closer to their centers. LIC contributes to a further improvement on both nodeAcc and rootAcc by combining the aforementioned two advantages. In contrast, compared to LIC, LISC provides only a marginal improvement by weighing the class centers because it only introduces small adjustments in the feature space. Further detailed comparison of these methods is presented in §5.3.

5.3 Visualization

LISC generates a more well-structured and discriminative representation space. Figure 4 shows a scatter plot of sentence and label embeddings, marked by dots and colored "×" respectively, and colored by classes. The distribution of the sampled examples in the figure is the same as the original dataset. Figures 4a - 4c show the representations extracted from bert-base, SCL, and LISC, respectively. We find that LISC generates a better representation than SCL by bringing clusters belonging to the same high-level classes closer to each other while simultaneously separating clusters of different classes. For instance, consider samples under the coarse-grained class "recreation" depicted in green. Initially, in Figure 4b, these sub-categories are widely dispersed. While in Figure 4c, the four sub-categories of "recreation" have become grouped closer to each other. This shows that penalizing the weights between classes with the class similarity matrix effectively guides the model to bring related sub-categories together. This can be interpreted to be a consequence of the ability of LISC to exploit dependencies among the classes, instead of considering each class independently as SCL does. In addition, the LISC also mitigates issues when there exist common themes where the corresponding label embeddings overlap one another.

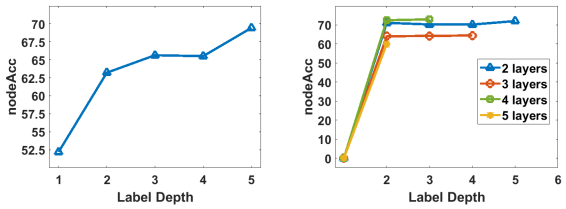
Method	IntraCluster ↓	InterCluster ↑
SCL	14.59	22.96
LI	14.32	23.66
LIUC	14.04	23.21
LIC	13.62	24.31
LISC	**13.52**	**24.48**

Table 3: Averaged inter- and intra-cluster L_2 distances on 20News, which measure the compactness and separation of clusters, respectively.

To quantitatively demonstrate the effectiveness of these methods, we calculate the average pairwise L_2 intra- and inter-cluster distances on 20News to measure the compactness of each cluster and distance between clusters as shown in Table 3. Smaller intra-cluster distance implies a more compact cluster. Meanwhile, the clusters are well-separated with a larger inter-cluster distance. Comparing SCL and LIUC, we can see that the additional instance-center-wise contrastive particularly improves cluster compactness by moving within-class examples closer to their centers. Comparing SCL to LI shows that the inter-cluster distance increases by applying class similarity to scale the temperature, leading to a more discriminative embedding space. LISC achieves the best performance among all variations by combining the aforementioned advantages. As a result, LISC facilitates clearer decision boundaries and improves the representation and organization in the embedding space.

5.4 Sensitivity to Different Label Hierarchies

Deeper hierarchical structures work better. To demonstrate the effect of hierarchy size, we assess how each leaf node label performs under different hierarchical structures. By manipulating the layers of the labels, we simulate different levels of granularity. To achieve this, we construct different label hierarchies with bottom-up levels ranging from 1-5 on 20News. The performance is always measured on the leaf nodes to make a fair comparison.



(a) (b)

Figure 5: Measure the sensitivity to different hierarchies on 20News in (a) nodeAcc with different bottom-up label hierarchies ranging from 1-5. (b) nodeAcc on labels grouped by different hierarchies.

We observe that the overall performance changes in response to different levels of label granularity, as shown in Figure 5a. A similar observation can be found in Figure 5b, which groups the performance based on the hierarchy of leaf nodes with depths ranging from 2-5. From Figure 5b, we notice that the model makes more precise predictions with more specific label information as the hierarchical depth increases. Besides, the proposed methods can also be applied to flat labels when the label depth is 1 given that we can leverage the label description as long as we have that prior knowledge. Thus, the model can better distinguish between closely related classes when provided with more detailed comprehensive labels.

6 Related Work

Learning Label Hierarchy Hierarchical text classification is a task involving assigning samples to specific labels (most commonly fine-grained levels) arranged in a structured hierarchy, which is typically represented as a tree or directed acyclic graph, where each node corresponds to a label (Pulijala and Gauch, 2004). Recent studies have suggested integrating the label structure into text features by encoding them with a label encoder. For instance, Chen et al. (2020a) embed the word and label hierarchies jointly in the hyperbolic space. Zhou et al. (2020) propose a hierarchy-aware global model to extract the label structural information. Zhang et al. (2022b) design a label-based attention module to extract information hierarchically from the labels on different levels. Wang et al. (2022) propose a network to embed label hierarchy to text encoder with contrastive learning. Chen et al. (2021a) propose a matching network to match labels and text at different abstraction levels. Other than these studies on network structure, Ge (2018) propose a hierarchical triplet loss, which is useful for finding hard negatives by hierarchically merging sibling branches. Recent work by (Zhang et al., 2022a) introduces a hierarchy-preserving loss, applying a hierarchical penalty to contrastive loss with the preservation of a hierarchical relationship between labels on images by using images under the same branch as positive pairs. Our LASCL, in contrast, exploits a small number of known labels and their hierarchical structure to improve the learning process. It differs from these works in constructing penalties from the hierarchical structure and exploiting it in the contrastive loss.

Contrastive Learning Self-supervised contrastive learning is a representation learning approach that maximizes agreement between augmented views of the same instance and pushes different instances far apart. Works on text data (Rethmeier and Augenstein, 2023) constructing various augmentations on text level (Wu et al., 2020; Xie et al., 2020; Wei and Zou, 2019; Giorgi et al., 2021), embedding level (Wei and Zou, 2019; Guo et al., 2019; Sun et al., 2020; Uddin et al., 2021), and via language models (Meng et al., 2021; Guo et al., 2019; Chuang et al., 2022), etc. SCL effectively learns meaningful representations and improves classification performance by combining supervised and contrastive learning advantages. It was initially introduced in SimCLR (Chen et al., 2020b). Other following works introduce novel insights to improve the representation learning such as MoCo (He et al., 2020), BYOL (Grill et al., 2020), and SwAV (Caron et al., 2020). SCL has also been applied to NLP tasks such as sentence classification (Chi et al., 2022), relation extraction (Li et al., 2022; Chen et al., 2021b) and text similarity (Zhang et al., 2021; Gao et al., 2021), where it has shown promising results in learning effective representations for text (Sedghamiz et al., 2021; Khosla et al., 2020; Chen et al., 2022).

Multi-label classification Multi-label text classification is to assign a subset of labels to a given text (Patel et al., 2022; Giunchiglia and Lukasiewicz, 2020). It acknowledges that a document can belong to more than one category simultaneously, and is especially useful when dealing with complex and diverse content that may cover multiple topics or themes. The modeling dependencies amongst labels in this work only consider assigning a single category to each sequence, and our future study is to extend this method to multi-label classification.

7 Conclusion

In this work, we propose LASCL to include information about the label hierarchy by introducing scaling to the SCL loss to penalize distances between negative example pairs using the class similarities constructed from the learned label feature representations. An additional instance-center-wise contrastive is introduced. These bring instances with similar semantics or belonging to the same high-level categories closer to each other, encourage each instance to become closer to its centers, and the underlying hierarchical structures can be encoded. A better-structured and discriminative feature space is generated by improving the intra-cluster compactness and inter-class separation. The learned labeled parameters can be directly applied as a nearest neighbor classifier without further tuning. Their effectiveness is demonstrated with experiments on three text classification datasets.

Limitations

Our proposed methods have some limitations, particularly when dealing with highly fine-grained label structures where most of the branches exhibit significant similarities. In this case, it is challenging to distinguish between label embedding similarities. Assigning weights to different classes may not be effective since the similarity scores $w_{cc'}$ are almost identical. This hinders the ability to accurately differentiate between classes and further impacts the performance. Another limitation comes from the common underlying issue of data. Bias can be learned by the model. To mitigate this, debias techniques can be employed to ensure fair and unbiased representation.

References

Wenbin An, Feng Tian, Ping Chen, Siliang Tang, Qinghua Zheng, and QianYing Wang. 2022. Fine-grained category discovery under coarse-grained supervision with hierarchical weighted self-contrastive learning. *EMNLP 2022*.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007+ ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings*, pages 722–735. Springer.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924.

Boli Chen, Xin Huang, Lin Xiao, Zixin Cai, and Liping Jing. 2020a. Hyperbolic interaction model for hierarchical multi-label classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7496–7503.

Haibin Chen, Qianli Ma, Zhenxi Lin, and Jiangyue Yan. 2021a. Hierarchy-aware label semantics matching network for hierarchical text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4370–4379.

Junfan Chen, Richong Zhang, Yongyi Mao, and Jie Xu. 2022. Contrastnet: A contrastive learning framework for few-shot text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10492–10500.

Tao Chen, Haizhou Shi, Siliang Tang, Zhigang Chen, Fei Wu, and Yueting Zhuang. 2021b. CIL: Contrastive instance learning framework for distantly supervised relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6191–6200, Online. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020b. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Jianfeng Chi, William Shand, Yaodong Yu, Kai-Wei Chang, Han Zhao, and Yuan Tian. 2022. Conditional supervised contrastive learning for fair text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2736–2756, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. 2022. DiffCSE: Difference-based contrastive learning for sentence embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4207–4218, Seattle, United States. Association for Computational Linguistics.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics.

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

Weifeng Ge. 2018. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–285.

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

Eleonora Giunchiglia and Thomas Lukasiewicz. 2020. Coherent hierarchical multi-label classification networks. *Advances in neural information processing systems*, 33:9662–9673.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*.

Xu Han, Pengfei Yu, Zhiyuan Liu, Maosong Sun, and Peng Li. 2018. Hierarchical relation extraction with coarse-to-fine grained attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2236–2245.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.

Kamran Kowsari, Donald E Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S Gerber, and Laura E Barnes. 2017. Hdltex: Hierarchical deep learning for text classification. In *2017 16th IEEE international conference on machine learning and applications (ICMLA)*, pages 364–371. IEEE.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine learning proceedings 1995*, pages 331–339. Elsevier.

Dongyang Li, Taolin Zhang, Nan Hu, Chengyu Wang, and Xiaofeng He. 2022. HiCLRE: A hierarchical contrastive learning framework for distantly supervised relation extraction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2567–2578, Dublin, Ireland. Association for Computational Linguistics.

Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. 2021. Prototypical contrastive learning of unsupervised representations. In *International Conference on Learning Representations*.

Nankai Lin, Guanqiu Qin, Gang Wang, Dong Zhou, and Aimin Yang. 2023. An effective deployment of contrastive learning in multi-label text classification. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8730–8744.

Wanqiu Long and Bonnie Webber. 2022. Facilitating contrastive learning of discourse relational senses by exploiting the hierarchy of sense relations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10704–10716, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mikołaj Małkiński and Jacek Mańdziuk. 2022. Multi-label contrastive learning for abstract visual reasoning. *IEEE Transactions on Neural Networks and Learning Systems*.

Yu Meng, Chenyan Xiong, Payal Bajaj, Paul Bennett, Jiawei Han, Xia Song, et al. 2021. Coco-lm: Correcting and contrasting text sequences for language model pretraining. *Advances in Neural Information Processing Systems*, 34:23102–23114.

Calvin Murdock, Zhen Li, Howard Zhou, and Tom Duerig. 2016. Blockout: Dynamic model selection for hierarchical deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2583–2591.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Dhruvesh Patel, Pavitra Dangati, Jay-Yoon Lee, Michael Boratko, and Andrew McCallum. 2022. Modeling label space interactions in multi-label classification using box embeddings. *ICLR 2022 Poster*.

Ashwin Pulijala and Susan Gauch. 2004. Hierarchical text classification. In *International Conference on Cybernetics and Information Technologies, Systems and Applications: CITSA*, volume 1, pages 257–262.

Nils Rethmeier and Isabelle Augenstein. 2023. A primer on contrastive pretraining in language processing: Methods, lessons learned, and perspectives. *ACM Computing Surveys*, 55(10):1–17.

Hooman Sedghamiz, Shivam Raval, Enrico Santus, Tuka Alhanai, and Mohammad Ghassemi. 2021. Supcl-seq: Supervised contrastive learning for downstream optimized sequence representations.

Robert R. Sokal and F. James Rohlf. 1962. The comparison of dendrograms by objective methods. *Taxon*, 11(2):33–40.

Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip S Yu, and Lifang He. 2020. Mixup-transformer: dynamic data augmentation for nlp tasks. *COLING*.

Varsha Suresh and Desmond Ong. 2021. Not all negatives are equal: Label-aware contrastive loss for fine-grained text classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4381–4394, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A F M Shahab Uddin, Mst. Sirazam Monira, Wheemyung Shin, TaeChoong Chung, and Sung-Ho Bae. 2021. Saliencymix: A saliency guided data augmentation strategy for better regularization. In *International Conference on Learning Representations*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Nakul Verma, Dhruv Mahajan, Sundararajan Sellamanickam, and Vinod Nair. 2012. Learning hierarchical similarity metrics. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2280–2287. IEEE.

Zihan Wang, Peiyi Wang, Lianzhe Huang, Xin Sun, and Houfeng Wang. 2022. Incorporating hierarchy into text encoder: a contrastive learning approach for hierarchical text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7109–7119, Dublin, Ireland. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268.

Siqi Zeng, Remi Tachet des Combes, and Han Zhao. 2023. Learning structured representations by embedding class hierarchy. In *The Eleventh International Conference on Learning Representations*.

Dejiao Zhang, Shang-Wen Li, Wei Xiao, Henghui Zhu, Ramesh Nallapati, Andrew O Arnold, and Bing Xiang. 2021. Pairwise supervised contrastive learning of sentence representations. *EMNLP 2021*.

Shu Zhang, Ran Xu, Caiming Xiong, and Chetan Ramaiah. 2022a. Use all the labels: A hierarchical multi-label contrastive learning framework. In *CVPR*.

Xinyi Zhang, Jiahao Xu, Charlie Soh, and Lihui Chen. 2022b. La-hcn: label-based attention for hierarchical multi-label text classification neural network. *Expert Systems with Applications*, 187:115922.

Jie Zhou, Chunping Ma, Dingkun Long, Guangwei Xu, Ning Ding, Haoyu Zhang, Pengjun Xie, and Gongshen Liu. 2020. Hierarchy-aware global model for hierarchical text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1106–1117.

A Appendix

A.1 LP with Label Embeddings

In the experiments of Section 5, we randomly initialized the parameters of the classifier. An alternative is to use the pretrained label-representative parameters as the linear head, and then to further train on the labeled dataset used in the linear probe. Results on 20NewsGroups are shown in Table 4. Comparing their performance to Table 2. Further tuning the label embedding matrix on labeled samples with cross-entropy loss impairs the performance with LI and LIUC. It achieves comparable or slightly better performance in terms of LISC and LIC.

Objective	nodeAcc	midAcc	rootAcc
LI	67.26	73.74	78.78
LIUC	64.42	68.08	78.45
LIC	68.99	72.90	80.75
LISC	69.15	76.00	81.40

Table 4: (%). LP by using label embeddings as an initialized classifier on 20NewsGroups.

A.2 Sensitivity on Different Label Templates

We explore the sensitivity of different label templates on 20NewsGroups as an example. Other than the template used in section §4, we also use the following templates

1. This sentence delivers $\{\text{label}_i\}$ news under the category of $\{\text{label}_i[L_1]\}$
2. Description of $\{\text{label}_i\}$ by generating a sentence from ChatGPT, the prompt given to ChatGPT is "Please generate a sentence to describe $\{\text{label}_i\}$ news."
3. $\{\text{label}_i\}$: description of $\{\text{label}_i\}$

In 2nd template, we use ChatGPT to generate a sentence description for each label. For instance, the description of "recreation,sport,hockey" is "In the latest recreation and sport news, hockey enthusiasts are buzzing with excitement as teams gear up for an intense season filled with thrilling matches and adrenaline-pumping action on the ice."

A.3 Comprehensive Few-Shot Cases Results

This section includes the full results in supplement to §5.1 shown in Table 6.

Templates	Objective	directly test			linear probe		
		nodeAcc	midAcc	rootAcc	nodeAcc	midAcc	rootAcc
1	LI	61.35	64.63	76.62	58.47	65.75	74.50
	LIUC	67.66	75.31	79.93	58.30	65.53	74 .44
	LIC	63.39	71.92	80.35	57.79	65.52	74.08
	LISC	67.34	75.66	79.43	57.78	65.44	74.16
2	LI	66.62	73.43	78.98	94.62	–	93.69
	LIUC	67.49	74.79	79.65	94.66	–	95.66
	LIC	65.45	73.88	80.02	94.25	–	95.35
	LISC	68.35	75.11	79.61	94.25	–	95.35
3	LI	65.43	72.29	78.52	66.88	73.62	79.13
	LIUC	67.69	74.88	80.24	94.66	–	95.66
	LIC	64.70	73.25	80.20	65.69	73.39	79.02
	LISC	67.90	75.00	79.49	94.25	–	95.35

Table 5: Results with different label templates on 20News.

Dataset	Objective	directly test			linear probe		
		nodeAcc	midAcc	rootAcc	nodeAcc	midAcc	rootAcc
1-shot							
20News	SCL	16.89	22.81	42.06	58.68	66.60	74.97
	LI	32.71	41.20	56.03	58.47	65.75	74.50
	LIUC	33.43	41.66	57.32	58.30	65.53	74 .44
	LIC	33.82	42.11	57.47	57.79	65.52	74.08
	LISC	33.30	40.96	56.47	57.78	65.44	74.16
WOS	SCL	0.32	–	12.22	34.39	–	52.05
	LI	0.70	–	14.43	49.94	–	66.08
	LIUC	0.41	–	13.30	49.33	–	65.18
	LIC	0.71	–	14.07	50.20	–	66.16
	LISC	0.70	–	14.47	50.69	–	66.23
DBPedia	SCL	0.52	–	22.95	95.50	–	95.56
	LI	1.45	–	20.9	94.62	–	93.69
	LIUC	1.42	–	21.33	94.66	–	95.66
	LIC	3.55	–	21.11	94.25	–	95.35
	LISC	3.58	–	20.26	94.25	–	95.35
100-shot							
20News	SCL	49.47	58.26	65.59	62.97	69.95	76.86
	LI	50.70	58.22	67.07	63.06	70.42	77.50
	LIUC	54.73	63.09	75.05	64.23	71.38	78.09
	LIC	63.52	70.83	78.21	63.21	70.17	76.95
	LISC	63.54	70.88	78.48	64.49	72.34	78.61
WOS	SCL	1.17	–	16.30	42.65	–	46.95
	LI	1.19	–	16.54	29.35	–	46.65
	LIUC	37.54	–	66.61	51.25	–	66.97
	LIC	59.59	–	72.70	61.14	–	73.25
	LISC	60.02	–	72.65	62.23	–	74.56
DBpedia	SCL	0.06	–	25.45	96.03	–	96.69
	LI	1.00	–	23.72	96.18	–	96.83
	LIUC	84.45	–	88.10	95.55	–	96.69
	LIC	93.13	–	94.48	95.80	–	96.61
	LISC	93.19	–	94.63	95.78	–	96.61

Table 6: Results on few-shot in supplement to §5.1.