Concept-Based Label Embedding via Dynamic Routing for Hierarchical Text Classification

Xuepeng Wang, Li Zhao, Bing Liu, Tao Chen, Feng Zhang, Di Wang
Tencent AI Platform Department, China
{woodswang,lilythzhao,andybliu}@tencent.com
{vitochen,jayzhang,diwang}@tencent.com

Abstract

Hierarchical Text Classification (HTC) is a challenging task that categorizes a textual description within a taxonomic hierarchy. Most of the existing methods focus on modeling the text. Recently, researchers attempt to model the class representations with some resources (e.g., external dictionaries). However, the concept shared among classes which is a kind of domain-specific and fine-grained information has been ignored in previous work. In this paper, we propose a novel concept-based label embedding method that can explicitly represent the concept and model the sharing mechanism among classes for the hierarchical text classification. Experimental results on two widely used datasets prove that the proposed model outperforms several state-of-the-art methods. We release our complementary resources (concepts and definitions of classes) for these two datasets to benefit the research on HTC.

1 Introduction

Text classification is a classical Natural Language Processing (NLP) task. In the real world, the text classification is usually cast as a hierarchical text classification (HTC) problem, such as patent collection (Tikk et al., 2005), web content collection (Dumais and Chen, 2000) and medical record coding (Cao et al., 2020). In these scenarios, the HTC task aims to categorize a textual description within a set of labels that are organized in a structured class hierarchy (Silla and Freitas, 2011). Lots of researchers devote their effort to investigate this challenging problem. They have proposed various HTC solutions, which are usually categorized into flat (Aly et al., 2019), local (Xu and Geng, 2019), global (Qiu et al., 2011) and combined approaches (Wehrmann et al., 2018).

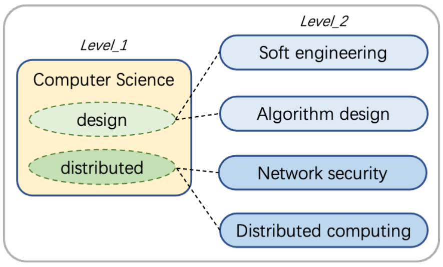

Figure 1: Concepts shared among classes in WOS.

In most of the previous HTC work, researchers mainly focus on modeling the text, the labels are simply represented as one-hot vectors (Zhu and Bain, 2017; Wehrmann et al., 2018). Actually, the one-hot vectors act as IDs without any semantic information. How to describe a class is also worthy of discussion. There is some work that embeds labels into a vector space which contains more semantic information. Compared with one-hot representations, label embeddings have advantages in capturing domain-specific information and importing external knowledge. In the field of text classification (includes the HTC task), researchers propose several forms of label embeddings to encode different kinds of information, such as 1) *anchor points* (Du et al., 2019), 2) *compatibility between labels and words* (Wang et al., 2018; Huang et al., 2019; Tang et al., 2015), 3) *taxonomic hierarchy* (Cao et al., 2020; Zhou et al., 2020) and 4) *external knowledge* (Rivas Rojas et al., 2020).

Although the external knowledge has been proven effective for HTC, it comes from a dictionary or knowledge base that humans constructed for entity definition, and it doesn't focus on the class explanations of a certain HTC task. In this sense, external knowledge is a type of domain-independent information. The taxonomic hierarchy encoding can capture the structural information of classes, which is a sort of domain-specific information for HTC. However, actually it only models the hypernym-hyponym relations in the class hierarchy. The process is implicit and difficult to

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 5010–5019
August 1–6, 2021. ©2021 Association for Computational Linguistics

be interpreted. Besides the structural connections between classes, we find that the information of concept shared between adjacent levels of classes is ignored by previous work. For instance, there is a parent node named “Sports” in a concrete class hierarchy (Qiu et al., 2011). Its subclasses “Surfing” and “Swimming” are “water” related sports. The subclasses “Basketball” and “Football” are “ball” related sports. The “water” and “ball” are a type of abstract concept included in the parent class “Sports” and can be shared by the subclasses. As shown in Figure 1, we have a similar observation in WOS (Kowsari et al., 2017), which is a widely used public dataset (details in our experiments). The concept “design” of the parent class “Computer Science” is shared by the child classes “Soft engineering” and “Algorithm design”. The concept “distributed” is shared by “Network security” and “Distributed computing”. The concept information can help to group the classes and measure the correlation intensity between parent and child classes. Compared with the information of node connections in the class hierarchy, the concept is more semantic and fine-grained, but rarely investigated. Although Qiu et al. (2011) have noticed the concept in HTC, they define the concept in a latent way and the process of represent learning is also implicit. Additionally, few of previous work investigates how to extract the concepts or model the sharing interactions among class nodes.

To further exploit the information of concept for HTC, we propose a novel *concept-based label embedding* method which can explicitly represent the concepts and model the sharing mechanism among classes. More specifically, we first construct a hierarchical attention-based framework which is proved to be effective by Wehrmann et al. (2018) and Huang et al. (2019). There is one concept-based classifier for each level. The prior level classification result (i.e. predicted soft label embedding) is fed into the next level. A label embedding attention mechanism is utilized to measure the compatibility between texts and classes. Then we design a concept sharing module in our model. It firstly extracts the concepts explicitly in the corpus and represents them in the form of embeddings. Inspired by the CapsNet (Sabour et al., 2017), we employ the dynamic routing mechanism. The iterative routing helps to share the information from the lower level to the higher level with the agreement in CapsNet. Taking into account the characters of HTC, we modify the dynamic routing mechanism for modeling the concepts sharing interactions among classes. In detail, we calculate the agreement between concepts and classes. An external knowledge source is taken as an initial reference of the child classes. Different from the full connections in CapsNet, we build routing only between the class and its own child classes to utilize the structured class hierarchy of HTC. Then the routing coefficients are iteratively refined by measuring the agreement between the parent class concepts embeddings and the child class embeddings. In this way, the module models the concept sharing process and outputs a novel label representation which is constructed by the concepts of parent classes. Finally, our hierarchical network adopts such label embeddings to represent the input document with an attention mechanism and makes a classification.

In summary, our major contributions include:

- This paper investigates the concept in HTC problem, which is a type of domain-specific information ignored by previous work. We summarize several kinds of existing label embeddings and propose a novel label representation: concept-based label embedding.
- We propose a hierarchical network to extract the concepts and model the sharing process via a modified dynamic routing algorithm. To our best knowledge, this is the first work that explores the concepts of the HTC problem in an explicit and interpretable way.
- The experimental results on two widely used datasets empirically demonstrate the effective performance of the proposed model.
- We complement the public datasets WOS (Kowsari et al., 2017) and DBpedia (Sinha et al., 2018) by exacting the hierarchy concept and annotating the classes with the definitions from Wikipedia. We release these complementary resources and the code of the proposed model for further use by the community[1].

2 Model

In this section, we detailedly introduce our model CLED (Figure 2). It is designed for hierarchical text classification with **C**oncept-based **L**abel

[1] https://github.com/wxpkanon/CLEDforHTC.git

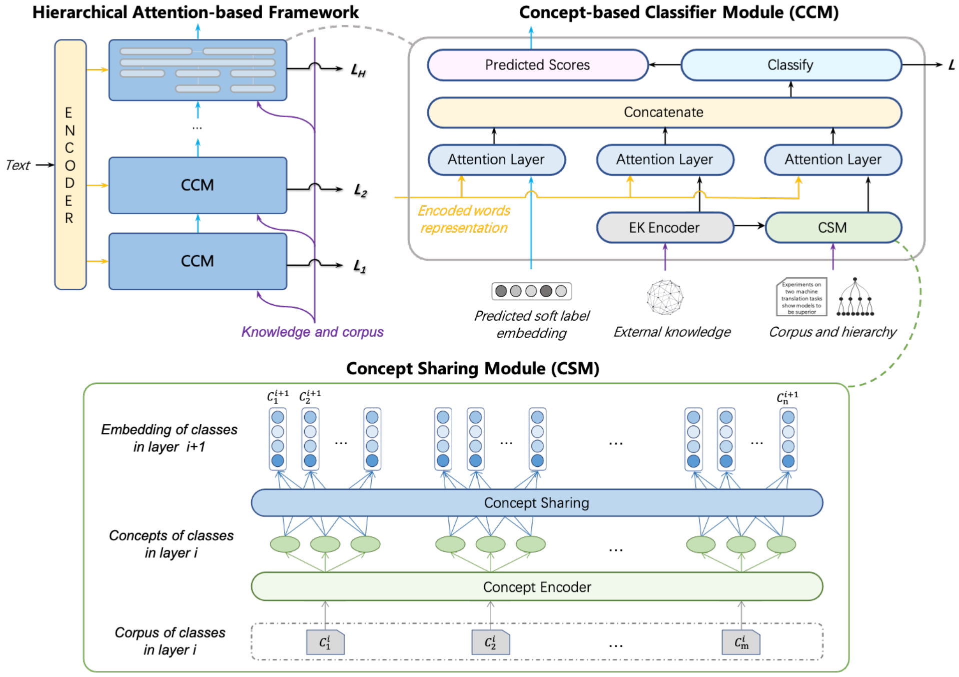
Hierarchical Attention-based Framework
Concept-based Classifier Module (CCM)
Text
ENCODER
CCM
CCM
...
L_H
L_2
L_1
Knowledge and corpus
Predicted Scores
Classify
L
Concatenate
Attention Layer
Attention Layer
Attention Layer
Encoded words representation
EK Encoder
CSM
Predicted soft label embedding
External knowledge
Experiments on two machine translation tasks show models to be superior
Corpus and hierarchy
Concept Sharing Module (CSM)
Embedding of classes in layer i+1
c_1^{i+1}
c_2^{i+1}
c_n^{i+1}
Concept Sharing
Concepts of classes in layer i
Concept Encoder
Corpus of classes in layer i
c_1^i
c_2^i
c_m^i

Figure 2: Illustration of our Concept-based Label Embedding via Dynamic routing (CLED) for HTC.

Embeddings via a modified **D**ynamic routing mechanism. Firstly, we construct a hierarchical attention-based framework. Then a concept sharing module is designed for extracting concepts and modeling the sharing mechanism among classes. The module learns a novel label representation with concepts. Finally, the model takes the concept-based label embeddings to categorize a textual description.

2.1 Hierarchical Attention-based Framework

In recent years, the hierarchical neural network has been proven effective for the HTC task by much work (Sinha et al., 2018; Wehrmann et al., 2018; Huang et al., 2019). We adopt it as the framework of our model.

Text Encoder We first map each document $d = (w_1, w_2, ..., w_{|d|})$ into a low dimensional word embedding space and denote it as $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{|d|})$. A CNN layer is used for extracting n-gram features. Then a bidirectional GRU layer extracts contextual features and represents the document as $\boldsymbol{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_{|d|})$.

Label Embedding Attention To measure the compatibility between labels and texts, we adopt the label embedding attention mechanism. Given a structured class hierarchy, we denote the label embeddings of the i-th level as $\boldsymbol{C} = (\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_{|l_i|})$, where $|l_i|$ is the number of classes in the i-th level. Then we calculate the cosine similarity matrix $\boldsymbol{G} \in \mathbb{R}^{|d| \times |l_i|}$ between words and labels via $g_{kj} = (\boldsymbol{s}_k^\top \boldsymbol{c}_j)/(\|\boldsymbol{s}_k\| \, \|\boldsymbol{c}_j\|)$ for the i-th level. Inspired by Wang et al. (2018) and Wang et al. (2019), we adopt convolutional filters $\boldsymbol{F}$ to measure the correlations $\boldsymbol{r}_p$ between the p-th phrase of length $2k + 1$ and the classes at i-th level, $\boldsymbol{r}_p = \mathrm{ReLU}(\boldsymbol{F} \otimes \boldsymbol{G}_{p-k:p+k} + \boldsymbol{b})$, where $\boldsymbol{b} \in \mathbb{R}^{|l_i|}$. We denote the largest correlation value of the p-th phrase with regard to the labels of i-th level as $t_p = \text{max-pooling}(\boldsymbol{r}_p)$. Then we get the label-to-text attention score $\boldsymbol{\alpha} \in \mathbb{R}^{|d|}$ by normalizing $\boldsymbol{t} \in \mathbb{R}^{|d|}$ with the SoftMax function. Finally, the document representation $\boldsymbol{d}_{att}$ can be obtained by averaging the word embeddings, weighted by label-to-text attention score: $\boldsymbol{d}_{att} = \sum_k^{|d|} \alpha_k \boldsymbol{s}_k$.

2.2 Concept Sharing Module (CSM)

Most of researchers focus on measuring the correlations of classes by modeling the structured class hierarchy. In fact, they only get the information of graphic connections. By contrast, the concepts are more semantic, fine-grained and interpretable, but have been ignored. To further exploit the concepts, we design a concept module to explicitly model the mechanism of sharing concepts among classes and measure the intensity of interactions.

Concepts Encoder Given the corpus of class c, we extract the keywords from the documents and take top-n ranked keywords as the concepts of class

Algorithm 1 Pseudo Code of Concepts Sharing via Dynamic Routing

```
Input: all the classes $c$ and their concepts $e$ in level $l$; all the classes in level $(l + 1)$
Output: $\boldsymbol{c}_j^{\mathrm{CL}}$: the concept-based label embedding of the class in level $(l + 1)$;
for each concept $i$ of a class $c$ in level $l$ and each of its child class $j$ in level $(l + 1)$: $b_{ij} \leftarrow 0$;
for $r$ iterations do
    for each concept $i$ of class $c$ in level $l$: $\boldsymbol{\beta}_i \leftarrow \mathrm{softmax}(\boldsymbol{b}_i)$;    ▷softmax computes Eq. 1
    for each child class $j$ of class $c$ in level $(l + 1)$: $\boldsymbol{v}_j \leftarrow \sum_i \beta_{ij}\boldsymbol{e}_i$;
    for each child class $j$ of class $c$ in level $(l + 1)$: $\boldsymbol{c}_j^{\mathrm{CL}} \leftarrow \mathrm{squash}(\boldsymbol{v}_j)$    ▷squash computes Eq. 4
    for each concept $i$ of class $c$ in level $l$ and each of its child class $j$ in level $(l + 1)$: $b_{ij} \leftarrow b_{ij} + \boldsymbol{e}_i \cdot \boldsymbol{c}_j^{\mathrm{CL}}$
end for
return $\boldsymbol{c}_j^{\mathrm{CL}}$
```

c. In the WOS dataset, every document is already annotated with several keywords. So we rank the keywords by term frequency within each class. For the DBpedia dataset, there is no annotated keyword available. We carry out the Chi-square (χ^2) statistical test, which has been widely accepted as a statistical hypothesis test to evaluate the dependency between words and classes (Barnard, 1992; Palomino et al., 2009; Kuang and Davison, 2017). The words are ranked by the χ^2 values. Having extracted concepts for each class, we represent them with word embeddings.

To further encode the concepts, we exploit two different ways and make a comparison in experiments. A simple and efficient way is to feed the concept embeddings into the sharing networks directly. Alternatively, we try the k-means clustering algorithm (Hartigan and Wong, 1979) in consideration of the similarity between concepts, then get the embeddings of cluster centers. The outputs (word embeddings or cluster centers) of concepts encoder are denoted as $\boldsymbol{E}^c = (\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_n)$ for class c.

Concepts Sharing via Dynamic Routing For the HTC task, we find that there are concepts of parent classes shared by their child classes. The semantically related classes share some concepts in common. The concepts describe a class in different views. We adopt the dynamic routing mechanism in the CapsNet (Sabour et al., 2017), which is effective for sharing the information from lower levels to higher levels. Considering the characters of HTC, we modify it to explicitly model the interactions among classes and quantitatively measure the intensity.

To utilize the taxonomic hierarchy, we build routing only between the class and its own child classes, which is different from the full connections in CapsNet. We take the coupling coefficients between concepts of a parent class and all its child classes as the intensities of the sharing interactions. The intensity (coupling coefficient) β_{ij} sums to 1 and is determined by a "routing softmax". The logit b_{ij} is the log prior probability that concept i of a parent class should be shared to its child class j in level l_n.

$$\beta_{ij} = \frac{\exp(b_{ij})}{\sum_k^{|l_n|} \exp(b_{ik})} \tag{1}$$

The logit b_{ij} is iteratively refined by adding with the agreement.

$$b_{ij} \leftarrow b_{ij} + \boldsymbol{e}_i \cdot \boldsymbol{c}_j^{\mathrm{CL}} \tag{2}$$

The agreement is the scalar product between the concept embedding $\boldsymbol{e}_i$ and the concept-based label embedding (CL) of the child class $\boldsymbol{c}_j^{\mathrm{CL}}$. The $\boldsymbol{v}_j$ is the intermediate label embedding of the child class, which is generated by weighting over all the concepts of its parent class.

$$\boldsymbol{v}_j = \sum_i \beta_{ij}\boldsymbol{e}_i \tag{3}$$

As Sabour et al. (2017) do in the CapsNet, we also use a non-linear "squashing" function which is effective in our experiments.

$$\boldsymbol{c}_j^{\mathrm{CL}} = \frac{\|\boldsymbol{v}_j\|^2}{1 + \|\boldsymbol{v}_j\|^2} \frac{\boldsymbol{v}_j}{\|\boldsymbol{v}_j\|} \tag{4}$$

Finally, we get the concept-based label embedding for class c_j by modeling the sharing mechanism. The new generated label embedding $\boldsymbol{c}_j^{\mathrm{CL}}$ is constructed with several concepts $\boldsymbol{e}_i$ in different views and affected in different intensities β_{ij}. Compared with randomly initializing $\boldsymbol{c}_j^{\mathrm{CL}}$, an external knowledge source is taken as an initial reference which is more effective in experiments. The procedures are illustrated in Algorithm 1.

2.3 Classification

We build a classifier for each class level. Let $\hat{\boldsymbol{y}}^{l_i}$ denote the predictions of the classes in i-th level.

$$\hat{\boldsymbol{y}}^{l_i} = \text{softmax}(\boldsymbol{W}_o \boldsymbol{m} + \boldsymbol{b}_o) \tag{5}$$

$$\boldsymbol{m} = \text{ReLU}(\boldsymbol{W}_m[\boldsymbol{d}^{\text{EK}}_{att}; \boldsymbol{d}^{\text{CL}}_{att}; \boldsymbol{d}^{\text{PRE}}_{att}] + \boldsymbol{b}_m) \tag{6}$$

where $\boldsymbol{W}_o, \boldsymbol{b}_o, \boldsymbol{W}_m, \boldsymbol{b}_m$ are learnable parameters and $[;]$ is the vector concatenating operator. The $\boldsymbol{d}^{\text{EK}}_{att}$ and $\boldsymbol{d}^{\text{CL}}_{att}$ are document representations weighted respectively by the label-to-text attention scores via external knowledge (EK) initialized label embeddings and concepts-based label embeddings (CL). To utilize the predictions in the (i-1)-th level, we feed the document represent $\boldsymbol{d}^{\text{PRE}}_{att}$ into the i-th level classifier. $\boldsymbol{d}^{\text{PRE}}_{att}$ is weighted by the attention scores of the predicted soft label embedding $\boldsymbol{c}^{\text{P}}$. $\boldsymbol{d}^{\text{PRE}}_{att} = \sum_k^{|d|} \alpha_k \boldsymbol{s}_k$, where $\alpha_k = (\boldsymbol{s}_k^\top \boldsymbol{c}^{\text{P}})/(\|\boldsymbol{s}_k\| \, \|\boldsymbol{c}^{\text{P}}\|)$, $\boldsymbol{c}^{\text{P}} = \sum_j^{|l_{i-1}|} \hat{y}_j^{l_{i-1}} \boldsymbol{c}_j^{\text{EK}}$ and $\boldsymbol{c}_j^{\text{EK}}$ is the label embedding represented by averaging word embeddings of class definition in external knowledge (EK encoder in Figure 2). We calculate the loss of classifier in i-th level as follows:

$$\mathcal{L}^{l_i} = \frac{1}{N} \sum_{n=1}^{N} \text{CE}(\boldsymbol{y}_n^{l_i}, \hat{\boldsymbol{y}}_n^{l_i}) \tag{7}$$

where $\boldsymbol{y}_n^{l_i}$ is the one-hot vector of ground truth label in the i-th level for document n and $\text{CE}(\cdot, \cdot)$ is the cross entropy between two probability vectors. We optimize the model parameters by minimize the overall loss function:

$$\mathcal{L} = \sum_{i=1}^{H} \mathcal{L}^{l_i} \tag{8}$$

where H is the total number of levels in the structured class hierarchy.

3 Experiments

3.1 Datasets

We evaluate our model on two widely used hierarchical text classification datasets: Web of Science (WOS; Kowsari et al. (2017)) and DBpedia (Sinha et al., 2018). The former includes published papers available from the Web of Science (Reuters, 2012). The latter is curated by Sinha et al. (2018) from DBpedia[2]. The general information of datasets is shown in Table 1. We complement these two datasets by extracting the hierarchy concepts and annotating the classes with the definitions from Wikipedia[3].

	WOS	DBpedia
# Classes in level 1	7	9
# Classes in level 2	134	70
# Classes in level 3	NA	219
# Documents	46,985	342,782
Train	28,479	278,408
Val	3,000	30,000
Test	15,506	34,374

Table 1: Statistics of WOS and DBpedia

3.2 Metrics and Parameter Settings

As the state-of-the-art methods do, we take the accuracy of each level and the overall accuracy as metrics. Hyper-parameters are tuned on a validation set by grid search. We take Stanford's publicly available GloVe 300-dimensional embeddings trained on 42 billion tokens from Common Crawl (Pennington et al., 2014) as initialization for word embeddings. The number of filters in CNN is 128 and the region size is $\{2, 3\}$. The number of hidden units in bi-GRU is 150. We set the maximum length of token inputs as 512. The rate of dropout is 0.5. The number of routing iterations is 3. We compare two different inputs of the sharing networks: 1) top 30 ranked concepts of each parent class as inputs; 2) 40 cluster centers generated by the k-means clustering algorithm on 1k concepts for each parent class. We train the parameters by the Adam Optimizer (Kingma and Ba, 2014) with an initial learning rate of 1e-3 and a batch size of 128.

3.3 Baselines

HDLTex Kowsari et al. (2017) prove that the hierarchical deep learning networks outperform the conventional approaches (Naïve Bayes or SVM).

HNATC Sinha et al. (2018) propose a Hierarchical Neural Attention-based Text Classifier. They build one classifier for each level and concatenate the predicted category embedding at (i-1)-th level with each of the encoder's outputs to calculate attention scores for i-th level.

[2] https://wiki.dbpedia.org/

[3] https://www.wikipedia.org/

Model	WOS			DBpedia			
	l_1	l_2	Overall	l_1	l_2	l_3	Overall
HDLTex	90.45	84.66	76.58	99.26	97.18	95.50	92.10
HNATC	89.32	82.42	77.46	99.21	96.03	95.32	93.72
HARNN	91.90	61.63	61.29	99.37	95.69	**95.71**	93.25
A-PNC-B	-	-	79.92	-	-	-	95.26
HiAGM-TP-LSTM	90.54	80.59	79.30	99.44	97.22	95.32	95.03
HiAGM-TP-GCN	90.78	80.79	79.34	99.43	97.18	95.29	94.85
HiAGM-LA-LSTM	90.20	80.09	78.28	99.40	97.14	95.12	94.64
HiAGM-LA-GCN	90.41	80.06	78.23	99.45	97.08	94.95	94.48
CLED	**93.40**	85.69	84.36	99.41	97.30	95.53	95.28
CLEDcluster	93.34	**86.19**	**85.13**	**99.46**	**97.36**	95.64	**95.39**

Table 2: Experimental results (accuracy, %) of our proposed model CLED and state-of-the-art methods. We evaluate the test set with the best model on the validation set. We run our model 5 times with different seeds and report the mean metrics. Improvements are statistically significant with p<0.01 based on the t-test. Note that Rivas Rojas et al. (2020) only report the overall accuracy for A-PNC-B.

HARNN Huang et al. (2019) propose a model called Hierarchical Attention-based Recurrent Neural Network with one classifier for each class level. They focus on modeling the dependencies among class levels and the text-label compatibility.

A-PNC-B Rivas Rojas et al. (2020) define the HTC as a sequence-to-sequence problem and propose a synthetic task of bottom-up-classification. They represent classes with external dictionaries. Their best combined strategy is Auxiliary task + Parent Node Conditioning (PNC) + Beam search.

HiAGM Zhou et al. (2020) propose a hierarchy-aware global model. They employ Tree-LSTM and hierarchy-GCN as the hierarchy encoder. Text feature Propagation (TP) and Label Attention (LA) are utilized for measuring the label-word compatibility. There are four HiAGM variants: TP-LSTM, TP-GCN, LA-LSTM, and LA-GCN.

3.4 Compared with State-of-the-art Methods

To illustrate the practical significance of our proposed model, we make comparisons with several competitive state-of-the-art methods. The results of experiments conducted on the public datasets are shown in Table 2. Most of the state-of-the-art methods referred to in Section 3.3 adopt a hierarchical attention-based network as their models' framework. Within their models, the hierarchical framework is effective in utilizing the classification results of the previous levels for the next levels. The label embedding attention mechanism helps to import external knowledge sources and the taxonomic hierarchy. On both of the two datasets, the state-of-the-art methods obtain competitive performance. With a similar framework, our model focuses on the concept-based label embedding and outperforms the other methods on both level and overall accuracy. The results indicate the effectiveness of the concepts among classes which have been ignored by previous work. The concept-based label embedding models related classes by the sharing mechanism with common concepts (visualizations in Section 3.6). The ablation comparisons are shown in Section 3.5.

The experimental results of the two variants of our model are also shown in Table 2. Compared with directly feeding the concepts into the sharing networks (CLED), the variant CLEDcluster performs slightly better. It indicates that cluster centers generated by the k-means algorithm are more informative and effective.

3.5 Ablation Experiments

To investigate the effectiveness of different parts in our model, we carry out ablation studies. The experiment results are shown in Table 3.

Effectiveness of Concept-based Label Embedding By comparing the results of CLED and the model without the learnt concept-based label embedding (w/o CL), we further confirm that the concepts shared among classes help to improve the performance.

Effectiveness of Dynamic Routing We remove the dynamic routing networks from the model CLED. Because there is no dynamic routing to share the concepts from the parent classes to their

Model	WOS			DBpedia			
	l_1	l_2	Overall	l_1	l_2	l_3	Overall
CLED	93.40	85.69	84.36	99.41	97.30	95.53	95.28
w/o CL	93.35	85.36	84.10	99.40	97.22	95.40	95.15
w/o EK	93.27	85.29	84.04	99.39	97.23	95.47	95.19
w/o PRE	93.34	85.33	84.03	99.39	97.18	95.35	95.05
w/o reference in CSM	93.30	85.45	84.17	99.40	97.18	95.45	95.15
w/o DR	93.29	85.41	84.23	99.36	97.23	95.38	95.12

Table 3: Ablation studies for different parts in our model.

child classes, it is an intuitive way to represent the label embeddings by averaging the word embeddings of the child classes' concepts. Specifically, there are top-30 ranked concepts for each parent class to share with their child classes. So for the model without dynamic routing (w/o DR), we represent the child class label embedding with the top-30 ranked concepts of each child class. Although the concepts of child classes are more fine-grained and informative than the concepts of parent classes, the model CLED with the dynamic routing networks to share the concepts among classes performs better. It indicates that modeling the sharing mechanism and learning to represent the child classes with common concepts are more effective.

Effectiveness of External Knowledge We take an external knowledge source as the initial reference of child classes in the concepts sharing module. When we remove the reference (w/o reference in CSM), the results are slightly worse on accuracy. It demonstrates that the external knowledge makes an efficient reference for the concept sharing.

Similar to the state-of-the-art methods, the external knowledge is also used individually as the representation of each class in our model. It helps to measure the compatibility between labels and texts via the attention mechanism. When we fully remove the external knowledge and initialize the label embeddings randomly (w/o EK), the performances are slightly worse than that with external knowledge (CLED). It indicates the effectiveness of external knowledge. Besides, the experiment which removes the predicted soft label embedding (w/o PRE) proves that, it is effective to utilize the predictions of previous level.

3.6 Visualizations of Concepts Sharing

In this paper, we explicitly investigate the concept sharing process. A concept sharing module is designed to model the mechanism of sharing concepts among classes and measure the intensity of interactions. The heat map of the learnt dynamic routing scores between the concepts of class "Computer Science" and its child classes is illustrated in Figure 3. The color changes from white to blue while the score increases. The score indicates the intensity between the concept and class in the sharing process. In Figure 3, we find that the concept "design" is shared by the classes "Soft engineering" and "Algorithm design". The concept "distributed" is shared by the classes "Network security" and "Distributed computing". The concept is shared by related classes.

We use t-SNE (Van der Maaten and Hinton, 2008) to visualize the concept embeddings of class "Computer Science" and the concept-based label embeddings of its child classes on a 2D map in Figure 4. The label embedding (red triangle) is constructed with the embeddings of concepts (blue dot). As shown, the class "Software engineering" is surrounded by the concepts "optimization" and "design". "Network security" is surrounded by "cloud", "machine" and "security". The class is described by several concepts in different views.

The visualizations in Figure 3 and 4 indicate that we successfully model the concept sharing mechanism in a semantic and explicit way.

4 Related Work

Hierarchical text classification with label embeddings Recently, researchers try to adopt the label embeddings in the hierarchical text classification task. Huang et al. (2019) propose hierarchical attention-based recurrent neural network (HARNN) by adopting label embeddings. Mao et al. (2019) propose to learn a label assignment policy via deep reinforcement learning with label embeddings. Peng et al. (2019) propose hierarchical taxonomy-aware and attentional graph RCNNs with label embeddings. Rivas Rojas et al. (2020)

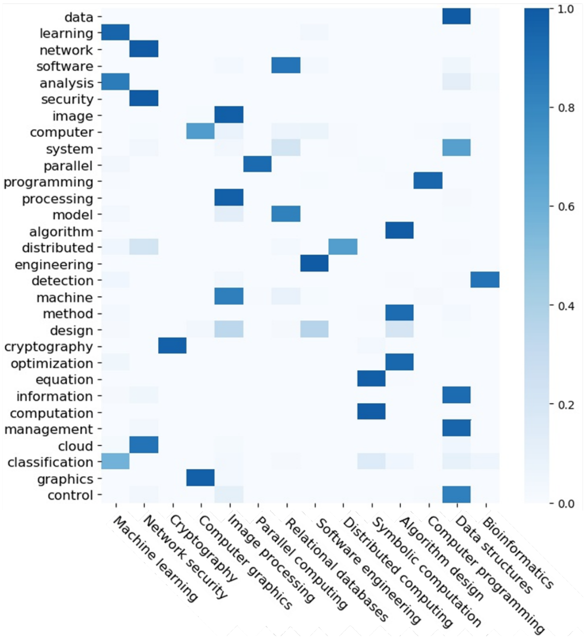

Figure 3: Dynamic routing scores between the concepts of class "Computer Science" (Y-axis) and its child classes (X-axis).

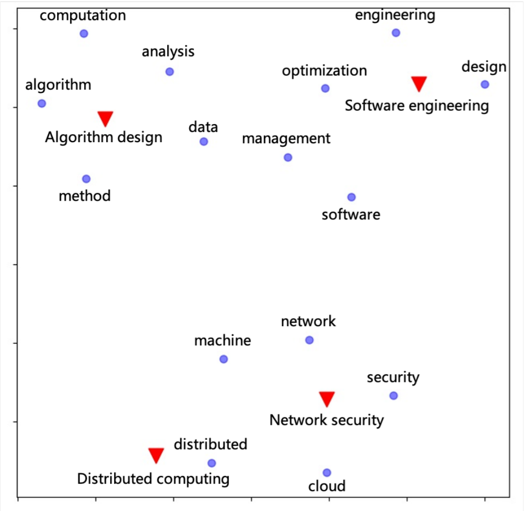

Figure 4: t-SNE plot of the concept embeddings of the class "Computer Science" and the concept-based label embeddings of its child classes.

define the HTC task as a sequence-to-sequence problem. Their label embedding is defined by external knowledge. For modeling label dependencies, Zhou et al. (2020) formulate the hierarchy as a directed graph and introduce hierarchy-aware structure encoders. Cao et al. (2020) and Chen et al. (2020a) exploit the hyperbolic representation for labels by encoding the taxonomic hierarchy.

Hierarchical text classification besides label embeddings According to the motivation of this work, we separate previous work with label embeddings from the HTC task and present it in the above paragraph. Besides, existing work is usually categorized into flat, local and global approaches (Silla and Freitas, 2011). The flat classification approach completely ignores the class hierarchy and only predicts classes at the leaf nodes (Aly et al., 2019). The local classification approaches could be grouped as a local classifier per node (LCN), a local classifier per parent node (LCPN) and a local classifier per level (LCL). The LCN approach train one binary classifier for each node of the hierarchy (Fagni and Sebastiani, 2007). Banerjee et al. (2019) apply transfer learning in LCN by fine-tuning the parent classifier for the child class. For the LCPN, a multi-class classifier for each parent node is trained to distinguish between its child nodes (Wu et al., 2005; Dumais and Chen, 2000). Xu and Geng (2019) investigate the correlation among labels by the label distribution as an LCPN approach. The LCL approach consists of training one multi-class classifier for each class level (Kowsari et al., 2017; Shimura et al., 2018). Zhu and Bain (2017) introduce a B-CNN model which outputs predictions corresponding to the hierarchical structure. Chen et al. (2020b) propose a multi-level learning to rank model with multi-level hinge loss margins. The global approach learns a global classification model about the whole class hierarchy (Cai and Hofmann, 2004; Gopal and Yang, 2013; Wing and Baldridge, 2014; Karn et al., 2017). Qiu et al. (2011) exploit the latent nodes in the taxonomic hierarchy with a global approach. For the need for a large amount of training data, a weakly-supervised global HTC method is proposed by Meng et al. (2019). Meta-learning is adopted by Wu et al. (2019) for HTC in a global way. In addition, there is some work combined with both local and global approach (Wehrmann et al., 2018). A local flat tree classifier is introduced by Peng et al. (2018) which utilizes the graph-CNN.

5 Conclusion

In this paper, we investigate the concept which is a kind of domain-specific and fine-grained information for the hierarchical text classification. We propose a novel concept-based label embedding model. Compared with several competitive state-of-the-art methods, the experimental results on two widely used datasets prove the effectiveness of our proposed model. The visualization of the concepts and the learnt concept-based label embeddings reveal the high interpretability of our model.

Acknowledgments

We sincerely thank Bingning Wang [4] for helpful discussions, and all reviewers and ACs for their insightful comments, time and efforts.

References

Rami Aly, Steffen Remus, and Chris Biemann. 2019. Hierarchical multi-label classification of text with capsule networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 323–330.

Siddhartha Banerjee, Cem Akkaya, Francisco Perez-Sorrosal, and Kostas Tsioutsiouliklis. 2019. Hierarchical transfer learning for multi-label text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6295–6300.

GA Barnard. 1992. Introduction to pearson (1900) on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. In *Breakthroughs in statistics*, pages 1–10. Springer.

Lijuan Cai and Thomas Hofmann. 2004. Hierarchical document categorization with support vector machines. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 78–87.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. 2020. Hypercore: Hyperbolic and co-graph representation for automatic icd coding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3105–3114.

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.

Tongfei Chen, Yunmo Chen, and Benjamin Van Durme. 2020b. Hierarchical entity typing via multi-level learning to rank. *arXiv preprint arXiv:2004.02286*.

Cunxiao Du, Zhaozheng Chen, Fuli Feng, Lei Zhu, Tian Gan, and Liqiang Nie. 2019. Explicit interaction model towards text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6359–6366.

Susan Dumais and Hao Chen. 2000. Hierarchical classification of web content. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 256–263.

Tiziano Fagni and Fabrizio Sebastiani. 2007. On the selection of negative examples for hierarchical text categorization. In *Proceedings of the 3rd Language & Technology Conference (LTC'07)*, pages 24–28. Citeseer.

Siddharth Gopal and Yiming Yang. 2013. Recursive regularization for large-scale classification with hierarchical and graphical dependencies. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 257–265.

John A Hartigan and Manchek A Wong. 1979. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108.

Wei Huang, Enhong Chen, Qi Liu, Yuying Chen, Zai Huang, Yang Liu, Zhou Zhao, Dan Zhang, and Shijin Wang. 2019. Hierarchical multi-label text classification: An attention-based recurrent network approach. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1051–1060.

Sanjeev Karn, Ulli Waltinger, and Hinrich Schütze. 2017. End-to-end trainable attentive decoder for hierarchical entity classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 752–758.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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.

Sicong Kuang and Brian D Davison. 2017. Learning word embeddings with chi-square weights for healthcare tweet classification. *Applied Sciences*, 7(8):846.

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

Yuning Mao, Jingjing Tian, Jiawei Han, and Xiang Ren. 2019. Hierarchical text classification with reinforced label assignment. 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 445–455.

[4] https://bingning.wang/research/aboutme

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2019. Weakly-supervised hierarchical text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6826–6833.

Marco A Palomino, Michael P Oakes, and Tom Wuytack. 2009. Automatic extraction of keywords for a multimedia search engine using the chi-square test. In *Proceedings of the 9th Dutch–Belgian information retrieval workshop (DIR 2009)*, pages 3–10.

Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. 2018. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 World Wide Web Conference*, pages 1063–1072.

Hao Peng, Jianxin Li, Senzhang Wang, Lihong Wang, Qiran Gong, Renyu Yang, Bo Li, Philip Yu, and Lifang He. 2019. Hierarchical taxonomy-aware and attentional graph capsule rcnns for large-scale multi-label text classification. *IEEE Transactions on Knowledge and Data Engineering*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Xipeng Qiu, Xuan-Jing Huang, Zhao Liu, and Jinlong Zhou. 2011. Hierarchical text classification with latent concepts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 598–602.

Thomson Reuters. 2012. Web of science.

Kervy Rivas Rojas, Gina Bustamante, Arturo Oncevay, and Marco Antonio Sobrevilla Cabezudo. 2020. Efficient strategies for hierarchical text classification: External knowledge and auxiliary tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2252–2257, Online.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866.

Kazuya Shimura, Jiyi Li, and Fumiyo Fukumoto. 2018. Hft-cnn: Learning hierarchical category structure for multi-label short text categorization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 811–816.

Carlos N Silla and Alex A Freitas. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72.

Koustuv Sinha, Yue Dong, Jackie Chi Kit Cheung, and Derek Ruths. 2018. A hierarchical neural attention-based text classifier. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 817–823.

Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1165–1174.

Domonkos Tikk, György Biró, and Jae Dong Yang. 2005. Experiment with a hierarchical text categorization method on wipo patent collections. In *Applied Research in Uncertainty Modeling and Analysis*, pages 283–302. Springer.

Bingning Wang, Ting Yao, Qi Zhang, Jingfang Xu, Zhixing Tian, Kang Liu, and Jun Zhao. 2019. Document gated reader for open-domain question answering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 85–94.

Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Joint embedding of words and labels for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2321–2331.

Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. 2018. Hierarchical multi-label classification networks. In *International Conference on Machine Learning*, pages 5075–5084.

Benjamin Wing and Jason Baldridge. 2014. Hierarchical discriminative classification for text-based geolocation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 336–348.

Feihong Wu, Jun Zhang, and Vasant Honavar. 2005. Learning classifiers using hierarchically structured class taxonomies. In *International symposium on abstraction, reformulation, and approximation*, pages 313–320. Springer.

Jiawei Wu, Wenhan Xiong, and William Yang Wang. 2019. Learning to learn and predict: A meta-learning approach for multi-label classification. *arXiv preprint arXiv:1909.04176*.

Changdong Xu and Xin Geng. 2019. Hierarchical classification based on label distribution learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5533–5540.

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.

Xinqi Zhu and Michael Bain. 2017. B-cnn: branch convolutional neural network for hierarchical classification. *arXiv preprint arXiv:1709.09890*.