Label-correction Capsule Network for Hierarchical Text Classification

Fei Zhao, Zhen Wu, Liang He, and Xin-Yu Dai

***Abstract*—Hierarchical Text Classification (HTC) aims to predict the category of a document in a given label hierarchy. Considering a parent-child relationship among labels at different levels, previous works mainly leverage the parent-level label information to guide the child-level classification and achieve promising results. However, they still suffer from two drawbacks: (1) insufficient for distinguishing similar labels at the same level; (2) fail to consider the error propagation problem caused by the incorrect parent-level predictions. For this reason, we first propose a hierarchical capsule network for the HTC task, due to the ability of capsules to distinguish similar categories. To ease the error propagation problem, we further devise two novel mechanisms in the proposed hierarchical capsule framework, i.e., *Label Injection* and *Label Re-Routing*, to enhance the tolerance of the model to the incorrect parent-level predictions. Experiments on two widely used datasets prove that our model achieves competitive performance. The ablation study further demonstrates the scalability of *Label Injection* and *Label Re-Routing*.**

***Index Terms*—Capsule Network, Text Classification, Attention Mechanism.**

I. Introduction

DOCUMENT classification is important to organize documents for retrieval and analysis. In recent years, document classification has drawn increasing attention with the rapid growth of the number of documents. In this work, we study the task of Hierarchical Text Classification (HTC), which aims to categorize documents into a set of labels that are organized in a class hierarchy. As shown in Table I, given a computer-related document, the first-level label is `computer science` and the second-level label is `computer graphics`, there is a parent-child relationship between two label concepts.

Considering the strong correlation among labels at different levels, existing methods mainly leverage the parent-level label information to guide the classification of the child-level. Following this idea, [1] combined deep neural networks in a top-down fashion where a separate neural network is built at each parent node to classify its children. Recently, [2] proposed an attention-based hierarchical framework, which adopts hard concatenation to combine the parent-level label vector with each token in the text representation to model label-text compatibility. [3] and [4] severally proposed a hierarchical recurrent neural network and a hierarchy-GCN structure to build the connection between labels at different levels. Furthermore, [5] turned the HTC task into a sequence-to-sequence problem, modeling the relationship among labels from the perspective of generation.

Fei Zhao, Zhen Wu, Liang He, and Xin-Yu Dai (Corresponding author) are with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China (e-mail: zhaof@smail.nju.edu.cn; wuz@nju.edu.cn; heliang@smail.nju.edu.cn; daixinyu@nju.edu.cn).

TABLE I
AN EXAMPLE OF THE HIERARCHICAL TEXT CLASSIFICATION TASK. AMONG THEM, COMPUTER GRAPHICS IS THE CHILDREN OF COMPUTER SCIENCE.

Document	3D digital visualization technology is a new research field along with the rapid development of computer technology. It consists of computer graphics, computer aided design...
Parent level	computer science
Child level	computer graphics

Although these studies have achieved great progress, there are still several limitations existing in the HTC task:

1) **Firstly**, each parent-level label contains numerous child-level labels, and as the level increase, the distinction between child-level labels at the same level becomes smaller and smaller, e.g. `computer vision` and `computer graphics`. These similar labels confuse the classifier seriously.
2) **Secondly**, they primarily focus on the performance improvement brought by the parent-level guidance but ignore the error propagation problem arising from the incorrect parent-level label in real-world scenarios. Take Table I for example, if the parent-level label `computer science` is predicted incorrectly, the error will accumulate for child-level predictions.

To address the above issues, we propose a novel **L**abel-correction **C**apsule **N**etwork (LCN) model for the HTC task. Specifically, inspired by the superiority of capsules [6], [7] in distinguishing similar features [8], [9], we design a hierarchical capsule framework, in which the labels at the same level corresponds to a group of competing capsules and each capsule is employed to classify one label. In this hierarchical capsule network, we further design two novel approaches to ease the error propagation problem. Firstly, we develop a *Label Injection* method to reduce errors between labels and text. This is because the existing works adopt the `hard` concatenation [2] of parent-level predictions and have a strong bias on text representations. In contrast, we add parent-level information on text representations in the style of `soft` weights and greatly reduce the bias. Secondly, we design a

dynamic *Label Re-Routing* mechanism to reduce the error propagation when parent-level labels and child-level labels are inconsistent. *Label Re-Routing* keeps parent-level guidance when two-level labels are consistent.

We conduct experiments on two widely used datasets. A series of experiments demonstrate that our LCN model achieves competitive results. Further analysis verifies the robustness and scalability of *Label Injection* and *Label Re-Routing*. We summarize the main contributions of this paper as follows:

1) To the best of our knowledge, we are the first to propose a new framework that takes into account both the case where the parent-level label predicts correctly and the case where the parent-level label predicts incorrectly.
2) We design two novel approaches to weaken the impact of incorrect parent-level labels on the child-level classification, which proved to be effective.
3) We explore and analyze the effect of pre-trained model BERT in the HTC tasks and achieve great improvement, which reflects the scalability of our methods.

The organization of this paper is as follows: In Section II, we discuss the differences between previous works and our proposed model LCN; In Section III, we introduce a novel model LCN, including details on the parent-level capsule module and child-level capsule module; Section IV contains the main experiment results on DBPedia and WOS datasets, which indicate that our LCN model enhances the tolerance of the HTC task; We validate the rationality of each component and provide in-depth analysis in Section V; Finally in Section VI, we give the conclusion of this work.

II. Related Work

In this section, we first review the existing studies on hierarchical text classification in detail. Then, considering that we use a hierarchical capsule network as the base model, we also present some representative works of the capsule network in the task of text classification.

A. Hierarchical Text Classification

With the rapid growth of the number of documents, the text classification task has drawn increasing attention. In this paper, we study a challenging text classification task, i.e., Hierarchical Text Classification (HTC). Generally, it can be categorized into two broad approaches: *global* and *local*. To be specific, the *global* approaches only use the last level of the entire taxonomy for training [1], [2], [5], [10]–[14]. For instance, [15] used a low dimensional vector to represented a text. [16], [17] first used an LSTM network to learn the hidden state, and than utilized max/mean pooling operations to predict the label. [18] proposed to exploit self-attention mechanism [19] instead of max/mean pooling operations to obtain sentence embedding. The *local* approaches creates a classifier at each level of the taxonomy [15]–[18], [20]–[22]. Recently, [1] took the documents retrieved from the parent-level label as inputs to the child-level in a pipeline manner. [2] proposed an end-to-end model HATC, which performs better than HDLTex at lower computation cost. [3] proposed a novel model called HARNN, which designed a hierarchical attention strategy to build the connection between labels among different levels. Similarly, [4] adopted TreeLSTM and hierarchy-GCN encoders to model label dependencies. Moreover, [5] addressed the HTC task from a different angle, turning it into a sequence-to-sequence problem and achieving good performance. However, all the above models mainly focus on the performance improvement brought by the parent-level guidance but ignore the error propagation problem arising from the incorrect parent-level label, which is a practical and challenging problem in the HTC task.

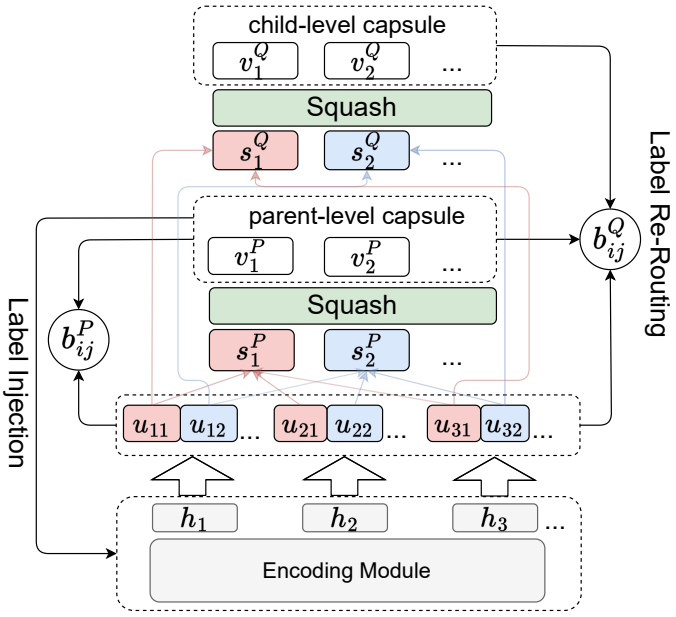

Fig. 1. Architecture of our Label-correction Capsule Network(LCN).

B. Capsule Network

Capsule network shows strong capabilities in traditional text classification tasks. [23] first attempted to use the capsule model to extract useful features for text classification. One interesting finding is that the capsule network has achieved competitive results when trained on a single-label text classification dataset and tested on a multi-label text classification task. [24] analyzed the weakness of CNN/RNN [25], [26] encoding text sequences and proposed a fixed-size encoding mechanism with a dynamic routing algorithm. [27] proposed a novel capsule-based architecture, which first designed a simple CNN-based module to obtain document representations, and then a capsule-based module with an attention mechanism to aggregate low-level document representations. In addition, capsule network is widely used in other fields, such as emotion detection [28], [29], sentiment analysis [30], relation extraction [31] and embedding creation for knowledge graph completion [32]. Although the capsule network has achieved good results in traditional classification tasks, its hierarchical structure has not been fully explored, which is crucial for hierarchical text classification tasks. This work takes advantage of the characteristics of the capsule network to fill this gap.

III. Label-correction Capsule Network

In this section, we introduce our Label-correction Capsule Network (LCN). We first present the task formalization of HTC, then illustrate the architecture of LCN in detail.

A. Task Formalization

Hierarchical Text Classification (HTC) aims to predict the category of a document in a given label hierarchy from top to bottom. Concretely, suppose we have a document that consists of n tokens $D = \{w_1, w_2, \cdots, w_n\}$ and category labels of m levels $C = \{c_1, \cdots, c_m\}$, $c_k \in \{c_1^{l_k}, \cdots, c_{s_k}^{l_k}\}$, where l_k and s_k refer to the k-th level of the class taxonomy and the number of classes in the k-th level. Note that the k-th level label is not only the parent-node of $(k+1)$ level label but also the children-node of $(k-1)$ level label. For the ease of the following descriptions, we assume that there are only two levels, the first level is the parent-level and the second level is the child-level, and they are denoted as c^P and c^Q.

B. Overview

Figure 1 shows the overall architecture of the Label-correction Capsule Network (LCN). It contains three main components: the encoding module (Sec. III-C), the parent-level capsule module (Sec. III-D), and the child-level capsule module (Sec. III-E). The encoding module is a typical bidirectional LSTM network [33] used for encoding the text. Between the encoding module and parent-level capsule module, we avoid the strong bias of hard concatenation and design the *Label Injection* mechanism, which utilizes parent-level predictions to generate `soft` weights to constraint text representation. The *Label Re-Routing* mechanism is introduced between the parent-level capsule and the child-level capsule. It keeps the parent-level guidance when two-level labels are consistent, but ignores the guidance when they are inconsistent.

C. Encoding Module

Given an input document $D = \{w_1, w_2, \cdots, w_n\}$ consist of n words, we first map each word into the corresponding word representations $\{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_n\}$ by using an embedding table $E_{emb} \in \mathbb{R}^{|V| \times D_W}$, where $|V|$ is the vocabulary size and D_W denotes the word embedding dimension. Then, we apply a bidirectional LSTM [33] to capture contextual representations $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_n\} \in \mathbb{R}^{n \times 2D_H}$, where D_H is the number of hidden units in BiLSTM.

D. Parent-level Capsule Module

Inspired by the superiority of capsule in distinguishing similar features, we exploit a capsule network for classification in the parent-level module, the procedures are shown in Algorithm 1. First of all, we define the extracted features $\mathbf{h}_i$ of token i as input capsules. Then, we transform each features vector $\mathbf{h}_i$ into the "vote vector" $\mathbf{u}_{j|i}^P$:

$$\mathbf{u}_{j|i}^P = \mathbf{W}_{ij}^P \mathbf{h}_i, \tag{1}$$

where $\mathbf{W}_{ij}^P \in \mathbb{R}^{D_P \times 2D_H}$ is a weight matrix, D_P is the dimension of the output capsule.

Algorithm 1 Dynamic routing-by-agreement

input: vote vectors $\mathbf{u}_{j|i}^P$ and number of iterations $iter_P$
output : output capsules $\mathbf{v}_j^P$
procedure DYNAMIC ROUTING($\mathbf{u}_{j|i}^P$, $iter_P$)
Initialize $b_{ij}^P \leftarrow 0$
for $iter_P$ iterations **do**
  for all input capsules : $c_{ij}^P = \text{softmax}(b_{ij}^P)$
  for all output capsules : $\mathbf{v}_j^P = \text{squash}(\sum_i^n c_{ij}\mathbf{u}_{j|i}^P)$
  for all input capsules and output capsules : $b_{ij}^P \leftarrow b_{ij}^P + \mathbf{u}_{j|i}^P \cdot \mathbf{v}_j^P$
end for
return $\mathbf{v}_j^P$ and b_{ij}^P

After that, we use a dynamic routing algorithm to assign "vote vector" $\mathbf{u}_{j|i}^P$ into the output capsules $\mathbf{v}_j^P$. The "squash" is a non-linear activation function:

$$\mathbf{v}_j^P = \text{squash}(\mathbf{s}_j^P) = \frac{||\mathbf{s}_j^P||^2}{1 + ||\mathbf{s}_j^P||^2} \frac{\mathbf{s}_j^P}{||\mathbf{s}_j^P||}, \tag{2}$$

where $\mathbf{s}_j^P$ is the inactivated output representations by the weighted sum of the "vote vectors":

$$\mathbf{s}_j^P = \sum_i^n c_{ij}^P \mathbf{u}_{j|i}^P, \tag{3}$$

where the coupling coefficients c_{ij}^P is computed by:

$$b_{ij}^P \leftarrow b_{ij}^P + \mathbf{u}_{j|i}^P \cdot \mathbf{v}_j^P, \tag{4}$$

$$c_{ij}^P = \text{softmax}(b_{ij}^P), \tag{5}$$

Finally, the loss function for the j-th class is calculated through the following equation:

$$\begin{aligned} \mathcal{L}_j^{parent} &= Y_j^P \max(0, m^+ - ||\mathbf{v}_j^P||)^2 \\ &+ \lambda(1 - Y_j^P)\max(0, ||\mathbf{v}_j^P|| - m^-). \end{aligned} \tag{6}$$

where Y_j^P = 1 if the document belongs to class $\mathbf{v}_j^P$, and Y_j^P = 0 if not. We simply set m^+, λ and m^- to 0.9, 0.5, 0.1, respectively. The total loss for the parent-level capsule module is $\mathcal{L}^{parent} = \sum_j^{s_P} L_j^{parent}$, with s_P being the number of parent-level classes.

E. Child-level Capsule Module

In real-world scenarios, the child-level capsule module needs to handle instances when the parent-level labels $\mathbf{v}_j^P$ in Equation 2 are predicted correctly and also when they are predicted incorrectly. To this end, we design two novel approaches to enhance the tolerance of the model to incorrectly-predicted parent-level labels. Concretely, our approaches are driven by the following research questions:

1) ***What** is the motivation for proposing the method?*
2) ***Why** can the approach ease the error propagation problem?*
3) ***How** can we implement the approach?*

1) Label Injection Method:

What: As mentioned before, the previous method from [2]

adopt a `hard` concatenation operation to combine the parent-level label vector with each token in the text representation. If the parent-level label is predicted incorrectly, the incorrect label would have a strong negative impact on each token. Even if the parent-level label predicts correctly, the interpretability of how the parent-level label information guides text is still weak due to the simple concatenation operation.

Why: In order to address the above problems, we propose a *Label Injection* approach, which uses `soft` weights generated by a gate mechanism to describe the relevance between the parent-level label and each token. This approach has two main advantages:

1) The incorrect label introduces noise to highly relevant tokens, while it does not introduce noise to tokens that are less relevant. This is in contrast to `hard` concatenation which adds a high amount of noise to each token. Consequently, our method greatly reduces the amount of overall noise between labels and text.
2) If the parent-level label gives the right prediction, these `soft` weights allow us to understand the importance of each token relative to the parent-level label, which enhances the interpretability.

How: We first use a gate mechanism to generate `soft` weight $\beta_i^Q \in [0,1]$ for each context word:

$$\beta_i^Q = \sigma(\mathbf{T}_Q \mathbf{v}_j^P + \mathbf{F}_Q \mathbf{h}_i + \mathbf{b}_Q), \tag{7}$$

where $\mathbf{T}_Q \in \mathbb{R}^{D_P \times 1}$ and $\mathbf{F}_Q \in \mathbb{R}^{2D_H \times 1}$ are weight matrix, $\mathbf{b}_Q$ is a bias. The generated `soft` weights β_i^Q merge the label information with its context. Then, we use these weights to route the capsules:

$$\mathbf{U}^Q = \mathbf{H} \odot \beta^Q, \tag{8}$$

where $\mathbf{U}^Q$ are the label-customized capsules and $\odot$ denotes element-wise multiplication. Finally, the child-level capsule module adopts the same dynamic routing-by-agreement algorithm as follows:

$$\mathbf{u}_{j|i}^Q = \mathbf{W}_{ij}^Q \mathbf{u}_i^Q, \tag{9}$$

$$\mathbf{v}_j^Q, b_{ij}^Q = \text{DYNAMIC ROUTING}(\mathbf{u}_{j|i}^Q, iter_Q), \tag{10}$$

where $\mathbf{W}_{ij}^Q \in \mathbb{R}^{D_Q \times D_P}$ is a weight matrix, D_P and D_Q are the dimensions of the input capsule and output capsule, $iter_Q$ is the number of iterations. We also use the max-margin loss for j-th class in the child-level capsule module:

$$\begin{aligned} \mathcal{L}_j^{child} &= Y_j^Q \max(0, m^+ - ||\mathbf{v}_j^Q||)^2 \\ &+ \lambda(1 - Y_j^Q)\max(0, ||\mathbf{v}_j^Q|| - m^-), \end{aligned} \tag{11}$$

The single loss for the child-level capsule module is $\mathcal{L}^{child} = \sum_{j=1}^{s_Q} L_j^{child}$, with s_Q being the number of classes in child-level. The final loss $\mathcal{L}$ of LCN is a linear combination of losses of different levels $\mathcal{L} = \mathcal{L}^{parent} + \mathcal{L}^{child}$.

2) Label Re-Routing Method:

What: In addition to *Label Injection*, we address the issue of error propagation from another perspective. Fundamentally, if the parent-level label predicts correctly, we hope it can help the child-level label make the right prediction. By contrast, if the parent-level label is predicted incorrectly, we hope that the child-level label will not be affected by the parent-level label. However, it is a large challenge to simultaneously model these two relationships between the parent-level label and the child-level label.

TABLE II
STATISTICS OF THE TWO DATASETS WOS AND DBPEDIA. THE NUMBER OF DOCUMENTS AND MEAN DOCUMENT LENGTH REPRESENT THE NUMBER OF DOCUMENTS AND MEAN LENGTH OF THE DOCUMENT RESPECTIVELY. IN ADDITION, WOS HAS TWO-LEVEL LABELS AND DBPEDIA HAS THREE-LEVEL LABELS.

	WOS	**DBPedia**
Number of documents	46,985	381,025
Mean document length	200.7	106.9
Classes in level 1	7	9
Classes in level 2	143	70
Classes in level 3	NA	219

Why: To overcome the above challenge, we propose a novel *Label Re-Routing* approach, which makes it possible for the parent-level label to be actively engaged in the dynamic routing of the child-level label. With the aid of dynamic routing, we can continuously adjust the relationship between the parent-level label and the child-level label. After multiple iterations, the model keeps the parent-level guidance when two-level labels are consistent, but ignores the guidance when they are inconsistent, which reduces errors between labels and labels to some degree.

How: When the child-level capsules update their routing weights, they not only consider the consistency with their own input capsules $\mathbf{u}_{j|i}^Q$, but also consider the consistency with the parent-level label $\mathbf{v}_j^P$. Formally, we re-update routing weights b_{ij}^Q in Equation 10 by:

$$b_{ij}^Q \leftarrow b_{ij}^Q + \mathbf{u}_{j|i}^Q \cdot \mathbf{v}_j^Q + \alpha \mathbf{v}_j^P \cdot \mathbf{v}_j^Q, \tag{12}$$

where α is the coefficient to model the relationship between the child-level label and the parent-level label.

IV. EXPERIMENTS

A. Datasets

We use two benchmark datasets for our experiments: WOS and DBpedia[1] [2]. The former is a hierarchical two-level taxonomy dataset that contains 46,985 documents, which provides the original text required for the training of deep neural models. The latter is a large-scale hierarchical three-level taxonomy dataset that contains 381,025 documents. The statistics for both datasets are presented in Table II. Following the previous work, we split both datasets into 90% for training (from which 10% were kept aside for validation) and 10%

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

TABLE III
DETAILS SETTING OF LCN.

Module	Hyper-parameters	Value
Encoding	word embedding dimension	300
	number of hidden units	200
	dropout	0.5
Parent-level Capsule	number of iterations	3
	dimension of the input capsule	32
	dimension of the output capsule	16
Child-level Capsule	number of iterations	3
	dimension of the input capsule	32
	dimension of the output capsule	16
	parameter α	0.6, 1.0
Trainer	learning rate	0.001
	batch size	64
	number of epochs	30
	optimizer	Adam

for testing. Finally, We report the average performance and standard deviation after being run five times.

B. Experiment Settings

For DBPedia and WOS datasets, we initialize word vectors from GloVe [34]. We also initialized all weight matrices and bias by a uniform distribution $U(-0.01, 0.01)$. The parameter α[2] in the *Label Re-Routing* is set to 0.6 and 1.0 on the WOS and DBPedia dataset, respectively. In addition, we adopt Adam optimizer [35] to update the parameters. We apply the Dropout [36] technology on the embedding layer with a probability of 0.5. Our models are implemented based on Tensorflow with an NVIDIA Tesla V100 GPU, and we select the best model in the validation set according to the accuracy score. The detail of hyper-parameters is shown in Table III.

C. Evaluation Metrics

For evaluating hierarchical models, we followed [2] and present the experimental results under two settings. The first setting is teacher-forcing, where we provide the true label of the parent-level to the child-level at training time, but provide the prediction results of the parent-level to the child-level during inference. Finally, we report the accuracy of the each level under teacher-forcing setting, such as l_1, l_2 and l_3. By contrast, the second setting provides the prediction results of the parent-level to the child-level during training, and also provides the prediction results of the parent-level to the child-level at inference time. The Overall score is the accuracy of the last level[3] in second setting. Considering that the true label of the parent-level cannot be obtained in real-world scenarios, this task mainly focuses on the Overall score.

D. Compared Methods

We divide baseline methods into two groups: flat-based models and hierarchical-based models.

[2]we analyze the effect of hyperparameter α in the Sec. V-D.

[3]We show the accuracy of different levels under the second setting in the Appendix

(I). The flat-based models only use the last level of the entire taxonomy for training:

- **FastText** [15] used a simple low-dimensional vector to represent the text.
- **BiLSTM + MLP + max/mean pooling** [16], [17] utilized an BiLSTM network and the average/max operation to extract useful features for classification.
- **Structured Self-attentive** [18] leveraged self-attention mechanism to highlight representative features in the text.

(II). The hierarchical-based models create a classifier at each level of the taxonomy for training:

- **HDLTex** [1] designed a hierarchical neural model, which takes the documents retrieved from the parent-level label as inputs to the child-level in a pipeline manner.
- **HATC** [2] proposed a simple attention framework, which adopts `hard` concatenation to combine the parent-level label vector with each token in the text representation to model label-text compatibility.
- **HARNN** [3] proposed a hierarchical recurrent neural network that design a hierarchical attention strategy to build the connection between labels among different levels in a top-down fashion.
- **HiAGM** [4] proposed a hierarchy-aware global model, which employs hierarchy-GCN and Tree-LSTM structure for modeling label dependencies.
- **Auxiliary task + PNC + Beam search** [5] addressed the HTC task from a different perspective, turning it into a sequence-to-sequence problem where the decoder generates labels of different levels in turn.

In addition, we also give a description of our base model and enhanced versions:

- **Hierarchical Capsule** concatenates the parent-level label vector with the child-level prediction vector to guide the classification of the child-level.
- **Hierarchical Capsule + Label Injection** utilizes the Label Injection to reduce the error propagation between the labels and text.
- **Hierarchical Capsule + Label Re-Routing** uses the Label Re-Routing to replace the concatenation operation between parent-level label vector and child-level prediction vector in Hierarchical Capsule to ease the error propagation when parent-level labels and child-level labels are inconsistent.
- **LCN** combines Label Injection and Label Re-Routing methods to obtain the final model.

E. Main Results

We display the main experiment results in Table IV. The first group contain some flat-based methods, the second group introduces existing hierarchical-based methods, our proposed model is in the last group.

From the first group, we observed that FastText performs very poorly because it loses a lot of useful information due to the simplicity of its design, the BiLSTM + MLP + Mean-pool method has achieved good performance on the Dbpedia dataset, but it does not perform well on the Wos dataset. In contrast, BiLSTM + MLP + Maxpool and Structured Self

TABLE IV
TEST ACCURACY ON THE WOS AND DBPEDIA DATASETS(%). NOTE THAT THE FLAT-BASED MODELS ONLY USE THE LAST LEVEL OF ENTIRE TAXONOMY FOR TRAINING. WE REPORT THE AVERAGE PERFORMANCE AND STANDARD DEVIATION AFTER BEING RUN FIVE TIMES, AND BEST RESULTS ARE IN BOLD ($p < 0.01$).

Methods	**DBPedia**				**WOS**		
	l_1	l_2	l_3	Overall	l_1	l_2	Overall
Flat models							
FastText	N/A	N/A	N/A	86.20	N/A	N/A	61.30
BiLSTM + MLP + Maxpool	N/A	N/A	N/A	94.20	N/A	N/A	77.69
BiLSTM + MLP + Meanpool	N/A	N/A	N/A	94.68	N/A	N/A	73.08
Structured Self Attention	N/A	N/A	N/A	94.04	N/A	N/A	77.40
Hierarchical models							
HDLTex	99.26	97.18	95.50	92.10	90.45	84.66	76.58
HATC	99.21	96.03	95.32	93.72	89.32	82.42	77.46
HARNN	99.19	96.06	95.55	93.86	90.41	82.65	77.98
HiAGM-GCN	99.13	96.63	96.70	94.91	90.76	83.03	79.36
HiAGM-TreeLSTM	99.19	97.12	96.89	95.02	90.62	83.82	79.61
Auxiliary task + PNC + Beam search	N/A	N/A	N/A	95.26	N/A	N/A	79.92
Ours models							
Hierarchical Capsule	99.25	96.58	96.18	94.00±0.111	90.43	83.93	79.81±0.144
LCN	99.29	97.31	97.10	**95.34**±0.088	90.39	85.05	**81.00**±0.092
BERT + Hierarchical Capsule	99.51	97.59	97.31	95.63±0.076	91.63	85.66	81.84±0.092
BERT + LCN	99.58	97.75	97.69	**96.06**±0.056	92.16	86.05	**82.73**±0.068

Attention models have a good performance on both datasets, which proves the importance of highlighting main features or selecting more meaningful words in text classification tasks.

From the second group, we can see that the accuracy of HATC and HDLTex is comparable to some flat classifiers. Compared with the HDLTex and HATC, HARNN and HiAGM achieve further improvement on the Dbpedia and Wos datasets, which demonstrates that HARNN and HiAGM make better use of the parent-level label information. In addition, we find that HiAGM performs better than HARNN since HiAGM adopts two effective hierarchical encoders, i.e., GCN and TreeLSTM. Overall, their results are still inferior to Auxiliary task + PNC + Beam search, which employs multiple efficient strategies to strengthen a baseline model, achieving competitive results. However, this approach relies on external dictionaries.

Hierarchical Capsule is our base model for HTC task. We observed that Hierarchical Capsule is significantly superior to BiLSTM models on the dataset WOS and slightly inferior to BiLSTM models on DBPedia. The differences may be attributed to that average/max pooling in BiLSTM models captures main features for classification on DBPedia and WOS, while easily losing some secondary features on the latter due to longer documents. After integrating both the *Label Injection* and *Label Re-Routing*, LCN obtains 1.19% and 1.34% improvement in accuracy on WOS and DBPedia datasets in contrast to the base model Hierarchical Capsules. From the statistics, the improvement of LCN on DBpedia is more significant, this may be because the DBpedia contains multiple levels, so the error propagation problem is more serious, which can give full play to the advantages of LCN. More importantly, LCN surpasses the strong baseline Auxiliary task + PNC + Beam search on both datasets. Although the improvement of LCN on DBPedia dataset is relatively small, considering that the DBPedia dataset is large-scale, our improvement is still considerable. These observations demonstrate that our model leverages the parent-level label information more effectively. Besides, we also replace the Glove+LSTM encoder in Hierarchical Capsule and LCN with the pre-trained model BERT to obtain BERT + Hierarchical Capsule and BERT + LCN. In contrast to Hierarchical Capsule, BERT + Hierarchical Capsule performs better on both datasets. Meanwhile, BERT + LCN achieves a new state-of-the-art performance. The reason behind this will be discussed in section V-E.

TABLE V
TEST ACCURACY OF DIFFERENT LEVELS ON THE WOS AND DBPEDIA DATASETS, Δ REPRESENTS THE DIFFERENCE BETWEEN THE PERFORMANCE OF HIERARCHICAL CAPSULE* AND HATC.

Methods	**DBPedia**			**WOS**	
	l_1	l_2	l_3	l_1	l_2
HATC	99.21	96.03	95.32	89.32	82.42
Hierarchical Capsule	99.25	96.58	96.18	90.43	83.93
Hierarchical Capsule*	99.23	96.38	96.02	89.93	83.48
Δ	+0.02	+0.35	+0.70	+0.61	+1.06

V. ABLATION STUDY

To further validate the origination of the improvement of LCN, we conduct ablation experiments and answer the

TABLE VI
ABLATION STUDY OVER TWO MAIN COMPONENTS.

Methods	DBPedia				WOS		
	l_1	l_2	l_3	Overall	l_1	l_2	Overall
Hierarchical Capsule	99.25	96.58	96.18	94.00±0.111	90.43	83.93	79.81±0.144
Hierarchical Capsule + Label Injection	99.24	97.12	96.70	94.67±0.067	90.23	84.40	80.62±0.086
Hierarchical Capsule + Label Re-Routing	99.28	97.15	96.80	94.89±0.076	90.35	84.81	80.72±0.083

TABLE VII
EXPERIMENT RESULTS OF INCORPORATING PARENT-LEVEL LABEL INFORMATION INTO TEXT REPRESENTATION BY ADDITION, CONCATENATION AND LABEL INJECTION.

Methods	DBPedia				WOS		
	l_1	l_2	l_3	Overall	l_1	l_2	Overall
Hierarchical Capsule	99.25	96.58	96.18	94.00±0.111	90.43	83.93	79.81±0.144
Hierarchical Capsule + Addition	99.20	96.74	96.51	94.24±0.083	90.23	84.02	80.01±0.157
Hierarchical Capsule + Concatenation	99.26	96.80	96.55	94.29±0.126	90.26	84.37	80.19±0.095
Hierarchical Capsule + Label Injection	99.24	97.12	96.70	94.67±0.067	90.23	84.40	80.62±0.086

following questions:

1) ***RQ1****: Does the Hierarchical Capsule can distinguish similar labels?*
2) ***RQ2****: Does the Label Injection can ease the error propagation between labels and text?*
3) ***RQ3****: Does the Label Re-Routing can alleviate the error propagation between labels and labels?*
4) ***RQ4****: Does the hyper-parameter α affect the performance of LCN?*
5) ***RQ5****: How much improvement can the BERT bring?*
6) ***RQ6****: What is the key advantage of our methods over other highly competitive approaches?*

A. Effect of Hierarchical Capsule (*RQ1*)

The main difference between HATC and Hierarchical Capsule is that the components (i.e., attention mechanism and capsule network) of their aggregation features are different. We illustrate the advantages of Hierarchical Capsule by comparing the results of the two models at each level. However, it is unreasonable to compare them directly because (1) HATC does not use pre-trained word vectors compared to Hierarchical Capsule; (2) Hierarchical Capsule and HATC adopt different concatenation methods when leveraging parent-level label vectors. For a fair comparison, we not only remove the Glove embedding from Hierarchical Capsule, but also use the same concatenation method as HATC, and finally named this variant as Hierarchical Capsule$^\bigstar$. From Table V, we can make a couple of observations:

1) The accuracy of Hierarchical Capsule$^\bigstar$ in each level is higher than HATC, which demonstrates that the capsule network is more effective than the attention mechanism in aggregating features to some extent.
2) As the level increases, there are more and more similar labels. We found that the performance of the Hierarchical Capsule$^\bigstar$ at each level is getting higher and higher than that of HATC, which verifies that the Hierarchical Capsule$^\bigstar$ is easier to distinguish the similar labels than HATC. This is in line with our motivation.

B. Effect of Label Injection (*RQ2*)

We explore three different ways of incorporating the parent-level label into the text representation. The easiest way is to adapt *Addition* or *Concatenation* operations to combine them together. Another way is our proposed *Label Injection* method. Table VII shows the experimental results, we can make a couple of summaries:

1) Compared to the base model Hierarchical Capsule, Hierarchical Capsule + Label Injection achieves competitive results on both datasets, which validates the rationale of incorporating parent-level label information with the text representation.
2) The performance of Hierarchical Capsule + Addition and Hierarchical Capsule + Concatenation is slightly higher than Hierarchical Capsule, but their results are much lower than Hierarchical Capsule + Label Injection, which further proves the validity of *Label Injection*.
3) As shown in Figure 2, we take the WOS dataset as an example. Compared with using its true label as the parent class, the accuracy of Hierarchical Capsule + Concatenation and Hierarchical Capsule + Label Injection drops by 4.18% and 3.78% respectively at the child level when using its predicted label as the parent class. Thus, we infer that the *Label Injection* method can reduce the noise between labels and text to some degree. This is consistent with our motivation.

C. Effect of Label Re-Routing (*RQ3*)

We also explore the effects of the *Label Re-Routing* method. Additionally, we calculate the percentage of child-level labels

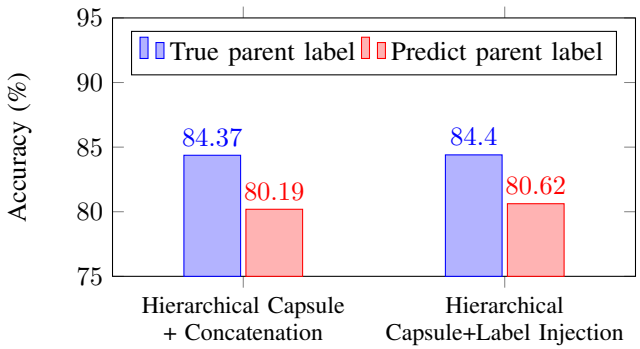

Fig. 2. The experiment results of Hierarchical Capsule + Concatenation and Hierarchical Capsule + Label Injection at child-level when using their true or predicted labels as the parent class.

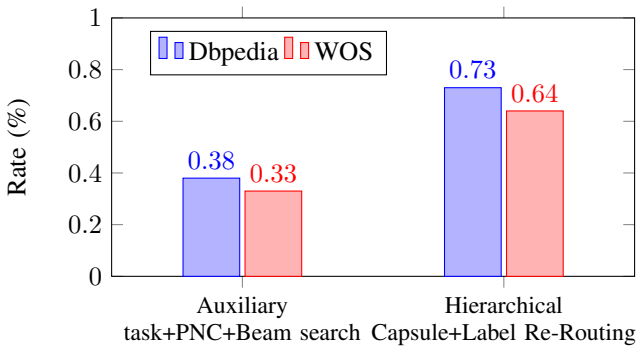

Fig. 4. The percentage of child-level labels that are predicted correctly when their parent-level labels are predicted incorrectly on Auxiliary task + PNC + Beam search and Hierarchical Capsule + Label Re-Routing.

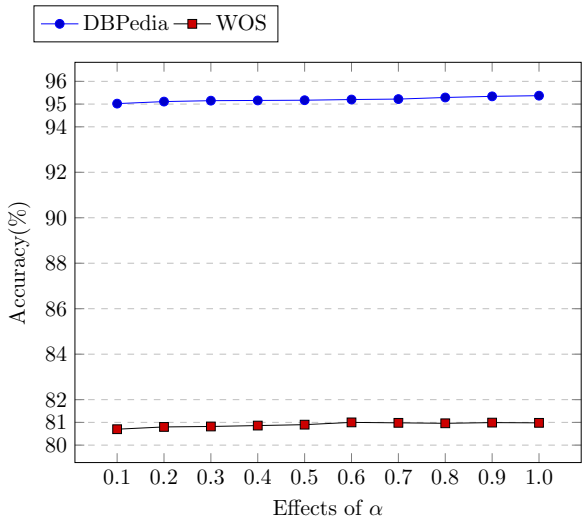

Fig. 3. The effect of different hyper-parameter α on LCN model.

that are predicted correctly when their parent-level labels are predicted incorrectly[4]. Based on Table IV and figure 4, we make the following summaries:

1) Compared to the base model Hierarchical Capsule, Hierarchical Capsule + Label Re-Routing performs better on both datasets. These results demonstrate that the relationship between the parent-level label and the child-level label is very important.
2) Hierarchical Capsule + Label Re-Routing is more effective than Hierarchical Capsule + Label Injection. One possible reason is that *Label Re-Routing* can continuously adjust the relationship between parent-level labels and child-level labels, which is beneficial for child-level labels to make correct predictions.
3) In contrast to the current state-of-the-art model Auxiliary task + PNC + Beam search, Hierarchical Capsule + Label Re-Routing has a higher percentage of child-level labels that are predicted correctly when their parent-level labels are predicted incorrectly (shown in figure 4), which indicates that *Label Re-Routing* can weaken the impact of the parent-level label errors on the child-level label.

[4]we calculate this percentage in the second setting.

TABLE VIII
THE HTC PERFORMANCE ON BERT-BASED MODEL. BEST RESULTS ARE IN BOLD ($p < 0.01$).

Methods	**DBPedia**	**WOS**
	Overall	
Flat models		
Bert	95.25±0.086	81.28±0.017
Bert + BiLSTM + MLP + Maxpool	95.34±0.126	81.35±0.108
Bert + Structured Self Attention	95.30±0.137	81.42±0.125
Hierarchical models		
Bert + HATC	95.42±0.041	81.77±0.096
Bert + Hierarchical Capsule	95.63±0.076	81.84±0.092
Bert + LCN	**96.06**±0.056	**82.73**±0.068

D. Effect of the Hyperparameter α (*RQ4*)

To evaluate the effect of hyper-parameter α on LCN model, we adjust the value of the α in Equation 12 and the interval is 0.1. Figure 3 displays the accuracy with different α on both datasets. Based on these results, we can make a couple of summaries:

1) As the α increases, LCN achieves stable performance on the WOS and Dbpedia datasets. It can be inferred from the results that *Label Injection* and *Label Re-Routing* have good robustness.
2) It is obvious that the curves on the WOS dataset show an overall upward trend when $\alpha < 0.6$, but become flat once $\alpha > 0.6$. By contrast, the performance of LCN has been showing an upward trend on the DBpedia dataset. Therefore, we finally set α to be 0.6 in the WOS dataset and 1.0 in the DBpedia dataset.

E. Effect of BERT (*RQ5*)

We also integrate the pre-trained model BERT [37] into the HTC to highlight the scalability of *Label Injection* and *Label Re-Routing*. To be specific, we replace the Glove+LSTM encoder with BERT and keep the other modules originally. Based on Table VIII, we can make following summaries:

PLOP	Document	Addition	Concatenation	Label Injection
event ✔	... the clock for elite racing cyclists held annually at marignylelozon in normandy france it was instituted in 1982 launched by a local cycling association the duo normand takes place on a road circuit of more than 54 km 34 mi every september	sportsteam ✘	sportsteam ✘	race ✔

Fig. 5. Visualization result of Hierarchical Capsule + Label Injection. PLOP is short for the prediction label of parent-level.

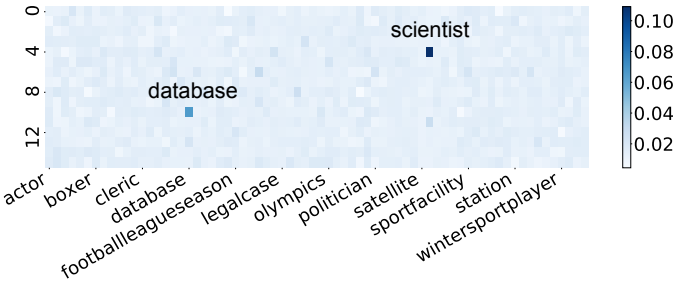

Fig. 6. Visualization result of Hierarchical Capsule.

Fig. 7. Visualization result of Hierarchical Capsule + Label Re-Routing.

1) In contrast to the LSTM-based model, the BERT-based model performs remarkably well on DBPedia and WOS datasets. A possible reason is that pre-trained models can provide abundant syntactic and semantic features.
2) Two BERT-based flat models perform slightly better than the pure BERT model, but their results are far inferior to our model BERT + LCN. This is consistent with our expectations since flat-based models do not leverage parent-level label information.
3) The results of Bert + HATC are inferior to Bert + Hierarchical Capsule. The reason behind this may be that Hierarchical Capsule is easier to distinguish similar labels than HATC.
4) BERT + LCN achieves further improvements on top of the strong base model BERT + Hierarchical Capsule, which further highlights the scalability of our methods.

*F. Case Study (**RQ6**)*

To highlight the advantage of our methods over base model, we select some samples from different datasets for a case study.

Part 1 This part aims to illustrate that Hierarchical Capsule + Label Injection has good interpretability:

In figure 5, we observed that the parent-level capsule module correctly predicts the label `event`. However, Hierarchical Capsule + Addition and Hierarchical Capsule + Concatenation both incorrectly predict the child-level label `sportsteam`. In contrast, Hierarchical Capsule + Label Injection focuses on some snippets which are closely related to the `event`, such as the time `every september`, the location `marignylelozon in normandy france` and the specific event `elite racing`, so the predicted label of the child-level capsule module is `race`.

Part 2 We present a representative example to show the advantage of Hierarchical Capsule + Label Re-Routing over Hierarchical Capsule:

As shown in figure 6[5] and figure 7, we visualize the coupling coefficients c_{ij} for Hierarchical Capsule and Hierarchical Capsule + Label Re-Routing, respectively. It is well-known that the parent-level capsule module incorrectly predicts the label `work`, and the true child-level label is `database`. By calculating the coupling coefficient c_{ij} after dynamic routing, we observed Hierarchical Capsule is highly coupled with the `database` and `scientist`. Consequently, Hierarchical Capsule incorrectly predicts the label `scientist`. In contrast, under the premise that the parent-level label is wrong, the dynamic routing process of Hierarchical Capsule + Label Re-Routing filters out the label `scientist`. Therefore, Hierarchical Capsule + Label Re-Routing can help the child-level label make the correct prediction.

VI. Conclusion and Future Work

In this paper, we propose a label-correction capsule network (LCN) for the challenging HTC task. Specifically, we first introduce a hierarchical capsule network that uses a group of competing capsules to classify at each level. Then we further design two novel methods, *Label Injection* and *Label Re-Routing*, to mitigate the key problem of the HTC task, namely error propagation caused by the incorrect parent-level labels. A series of experiments indicate that the LCN model outperforms all baseline methods. Further analysis validates

[5]we only show some labels on the horizontal axis.

TABLE IX
TEST ACCURACY ON THE WOS AND DBPEDIA DATASETS. **NOTE THAT** HERE WE USE THE PREDICTED LABEL AS THE PARENT DURING TRAINING.

Methods	**Dbpedia**			**WOS**	
	L_1	L_2	L_3	L_1	L_2
Hierarchical Capsule	99.07	96.07	**94.00**±0.111	90.23	**79.81**±0.144
Hierarchical Capsule + Label Injection	99.22	96.63	**94.67**±0.067	90.40	**80.62**±0.086
Hierarchical Capsule + Label Re-Routing	99.26	96.60	**94.89**±0.076	90.44	**80.72**±0.083
LCN	99.29	97.05	**95.34**±0.088	90.78	**81.00**±0.092

TABLE X
THE PERFORMANCE OF THE ATTENTION-BASED MODEL HATC AND THE ENHANCED VERSION.

Methods	**DBPedia**				**WOS**		
	l_1	l_2	l_3	Overall	l_1	l_2	Overall
HATC	99.21	96.03	95.32	93.72	89.32	82.42	77.46
HATC + Label Injection	99.23	97.05	96.65	94.63±0.094	89.76	82.76	78.61±0.055
HATC + Label Re-Routing*	99.26	97.14	96.73	94.77±0.084	89.91	82.82	78.74±0.098
HATC + Label Injection + Label Re-Routing*	99.27	97.30	97.02	95.11±0.110	90.19	83.18	79.38±0.057

the robustness and scalability of *Label Injection* and *Label Re-Routing*. Since our methods do not leverage the grammatical or syntactic information in the pre-trained model BERT, it is inevitable to consider how to leverage them effectively, which is one of the research plans in future work.

ACKNOWLEDGMENTS

This work is supported by National Key R&D Program of China (2018YFB1005102) and National Science Foundation of China (No. 61976114 and No.61936012).

APPENDIX

THE OVERALL SCORE

In Table IX, we display the experimental results of different levels during inference when the classifier uses its own prediction results as the parent class at training time. The result of the last level is the Overall score. We can observe that the result of each level is inferior to teacher-forcing results, it is reasonable since we don't use true labels during training.

UNIVERSAL ANALYSIS

Besides, we also extend our approaches to the attention-based model HATC. It should be noted that Label Re-Routing relies on the dynamic routing process of the capsule network, so we cannot directly extend it to the attention-based method. To bridge this gap, we propose a variant of Label Re-Routing named Label Re-Routing*, i.e., we add the relationship between the parent-level label $\mathbf{v}_j^P$ and the child-level label $\mathbf{v}_j^Q$ in Equation 12 to the child-level label prediction instead of the dynamic routing process, the experimental results are shown in Table X. Based on these results, we can observe that: (1) Compared to the baseline HATC, HATC + Label Injection and HATC + Label Re-Routing* achieve competitive results, which demonstrate that our methods have good universality; (2) After integrating two methods at the same time, the performance of HATC + Label Injection + Label Re-Routing* obtain further improvements, which demonstrate that Label Injection and Label Re-Routing* improve the performance of the HTC task from different perspectives. Overall, these results prove that our method has good robustness and universality.

REFERENCES

[1] K. Kowsari, D. E. Brown, M. Heidarysafa, K. J. Meimandi, M. S. Gerber, and L. E. Barnes, "Hdltex: Hierarchical deep learning for text classification," in *16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017, Cancun, Mexico, December 18-21, 2017*. IEEE, 2017, pp. 364–371.

[2] K. Sinha, Y. Dong, J. C. K. Cheung, and D. Ruths, "A hierarchical neural attention-based text classifier," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Association for Computational Linguistics, 2018, pp. 817–823.

[3] W. Huang, E. Chen, Q. Liu, Y. Chen, Z. Huang, Y. Liu, Z. Zhao, D. Zhang, and S. Wang, "Hierarchical multi-label text classification: An attention-based recurrent network approach," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*. ACM, 2019, pp. 1051–1060.

[4] J. Zhou, C. Ma, D. Long, G. Xu, N. Ding, H. Zhang, P. Xie, and G. Liu, "Hierarchy-aware global model for hierarchical text classification," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics, 2020, pp. 1106–1117.

[5] K. R. Rojas, G. Bustamante, A. Oncevay, and M. A. S. Cabezudo, "Efficient strategies for hierarchical text classification: External knowledge and auxiliary tasks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics, 2020, pp. 2252–2257.

[6] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *Artificial Neural Networks and Machine Learning - ICANN 2011 - 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I*, ser. Lecture Notes in Computer Science, vol. 6791. Springer, 2011, pp. 44–51.

[7] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017, pp. 3856–3866.

[8] X. Zhang, P. Li, W. Jia, and H. Zhao, "Multi-labeled relation extraction with attentive capsule network," in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 2019, pp. 7484–7491.

[9] C. Du, H. Sun, J. Wang, Q. Qi, J. Liao, T. Xu, and M. Liu, "Capsule network with interactive attention for aspect-level sentiment classification," 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 2019, Hong Kong, China, November 3-7, 2019*. Association for Computational Linguistics, 2019, pp. 5488–5497.

[10] S. Liu, M. Dong, H. Zhang, R. Li, and Z. Shi, "An approach of multi-hierarchy text classification," in *2001 International Conferences on Info-Tech and Info-Net. Proceedings (Cat. No. 01EX479)*, vol. 3. IEEE, 2001, pp. 95–100.

[11] M. J. Quinn and M. L. Laier, "Method and apparatus for fast lookup of related classification entities in a tree-ordered classification hierarchy," Apr. 18 2006, uS Patent 7,032,072.

[12] C. Vens, J. Struyf, L. Schietgat, S. Dzeroski, and H. Blockeel, "Decision trees for hierarchical multi-label classification," *Mach. Learn.*, vol. 73, no. 2, pp. 185–214, 2008.

[13] S. Gopal and Y. Yang, "Recursive regularization for large-scale classification with hierarchical and graphical dependencies," in *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*. ACM, 2013, pp. 257–265.

[14] J. Wehrmann, R. Cerri, and R. C. Barros, "Hierarchical multi-label classification networks," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 2018, pp. 5225–5234.

[15] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*. Association for Computational Linguistics, 2017, pp. 427–431.

[16] R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, ser. ACM International Conference Proceeding Series, vol. 307. ACM, 2008, pp. 160–167.

[17] J. Y. Lee and F. Dernoncourt, "Sequential short-text classification with recurrent and convolutional neural networks," in *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*. The Association for Computational Linguistics, 2016, pp. 515–520.

[18] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[19] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[20] C. N. S. Jr. and A. A. Freitas, "A global-model naive bayes approach to the hierarchical prediction of protein functions," in *ICDM 2009, The Ninth IEEE International Conference on Data Mining, Miami, Florida, USA, 6-9 December 2009*. IEEE Computer Society, 2009, pp. 992–997.

[21] R. Aly, S. Remus, and C. Biemann, "Hierarchical multi-label classification of text with capsule networks," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 2: Student Research Workshop*. Association for Computational Linguistics, 2019, pp. 323–330.

[22] X. Qiu, X. Huang, Z. Liu, and J. Zhou, "Hierarchical text classification with latent concepts," in *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - Short Papers*. The Association for Computer Linguistics, 2011, pp. 598–602.

[23] W. Zhao, J. Ye, M. Yang, Z. Lei, S. Zhang, and Z. Zhao, "Investigating capsule networks with dynamic routing for text classification," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 3110–3119.

[24] J. Gong, X. Qiu, S. Wang, and X. Huang, "Information aggregation via dynamic routing for sequence encoding," in *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*. Association for Computational Linguistics, 2018, pp. 2742–2752. [Online]. Available: https://aclanthology.org/C18-1232/

[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[26] X. Zhang, J. J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 2015, pp. 649–657.

[27] W. Zheng, Z. Zheng, H. Wan, and C. Chen, "Dynamically route hierarchical structure representation to attentive capsule for text classification," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*. ijcai.org, 2019, pp. 5464–5470. [Online]. Available: https://doi.org/10.24963/ijcai.2019/759

[28] S. Srivastava, P. Khurana, and V. Tewari, "Identifying aggression and toxicity in comments using capsule network," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 98–105.

[29] P. Rathnayaka, S. Abeysinghe, C. Samarajeewa, I. Manchanayake, and M. Walpola, "Sentylic at IEST 2018: Gated recurrent neural network and capsule network based approach for implicit emotion detection," in *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 254–259.

[30] Y. Wang, A. Sun, J. Han, Y. Liu, and X. Zhu, "Sentiment analysis by capsules," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1165–1174.

[31] N. Zhang, S. Deng, Z. Sun, X. Chen, W. Zhang, and H. Chen, "Attention-based capsule networks with dynamic routing for relation extraction," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 986–992.

[32] D. Q. Nguyen, T. Vu, T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A capsule network-based embedding model for knowledge graph completion and search personalization," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 2180–2189.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014, pp. 1532–1543.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[36] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012.

[37] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1*, 2019, pp. 4171–4186.

Fei Zhao is currently a PhD Candidate in natural language processing at Nanjing University. He received the Master degree from the Department of Computer Science and Technology, Nanjing University, in 2021. His research interests include text classification and sentiment analysis.

Zhen Wu is a research Associate Researcher of the School of Artificial Intelligence at Nanjing University, China. He received his Ph.D. degree in the Department of Computer Science & Technology at Nanjing University in 2021, and received his B.Eng. degree from Nanjing University of Science and Technology in 2016. His research interests include sentiment analysis, opinion mining, sentiment generation, transfer learning, and deep learning.

Liang He received the Master degree from the Department of Computer Science and Technology, Nanjing University, in 2008. He was S/W Architect and Principal Engineer of Samsung Electronics during 2008 2018, leading the Data Analytics Lab and AI Architect Committee in China R&D Center. He is currently a PhD Candidate in natural language processing at Nanjing University. His research interests include relation extraction and knowledge graph.

Xin-Yu Dai received the B.Eng. and Ph.D. degrees from the Department of Computer Science and Technology, Nanjing University, in 1999 and 2005, respectively. He has been on leave from Nanjing University from August 2010 to September 2011 to visit EECS Department and Statistics Department at UC Berkeley. He is currently a Professor with the School of Artificial Intelligence at Nanjing University. His research interests include natural language processing and knowledge engineering.