Utilizing Local Hierarchy with Adversarial Training for Hierarchical Text Classification

LREC-COLING 2024

Zihan Wang, Peiyi Wang, Houfeng Wang*

National Key Laboratory of Multimedia Information Processing, Peking University, China

*Corresponding author.

Overview

What is Hierarchical Text Classification?

- Text classification with label hierarchy.
- A subtask of multi-label classification. Label set $\mathcal{Y}$ contains k different candidates. Each input has a ground-truth label set $Y \subset \mathcal{Y}$.
- Taxonomic hierarchy (global hierarchy) $G = (\mathcal{Y}, E)$ is a tree. Ground-truth label set (local hierarchy) Y contains one or more paths.

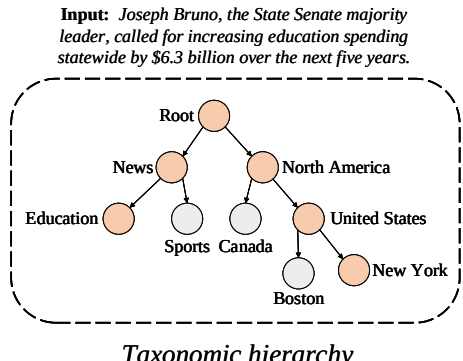

Figure 1: Example of a taxonomic hierarchy.

Background

- Nearly all of the recent HTC works modeled the label hierarchy with a graph encoder. [1]–[4].
- These works mainly focus on the constant global hierarchy but ignore the subgraph corresponding to each input text.
- Although a few works incorporate this local hierarchy as flat [5] or as a sequence [6], full hierarchy information is lost.

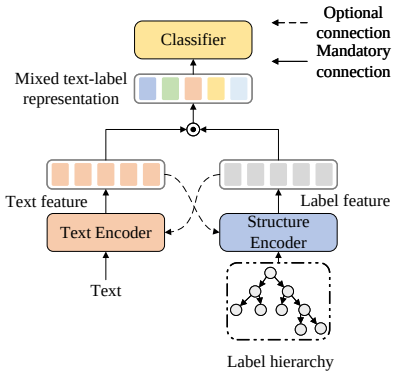

Figure 2: An abstract of recent HTC models.

Introduction

We attempt to directly encode the local hierarchy as a graph through adversarial training.

- We view the local hierarchy as a "real" input and it can generate an oracle representation that can perform perfect classification.
- We then train the mixed text-label representation and the oracle representation adversarially.
- After training, the raw representation should have a similar classification performance as the oracle representation.

Hierarchy-aware Adversarial Framework (HiAdv)

Our framework contains:

- A generator
- An encoder
- A discriminator

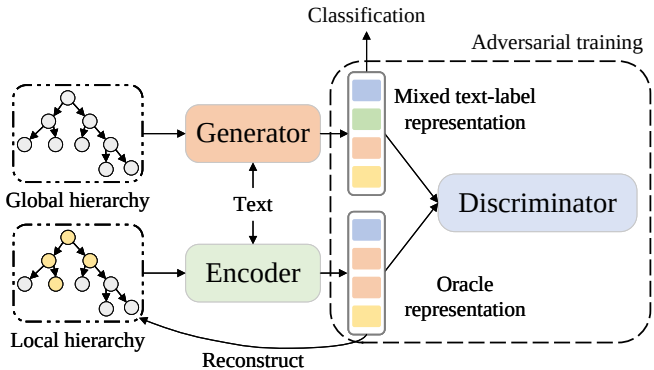

Figure 3: The architecture of HiAdv during training.

HiAdv: Generator

- The generator aims to generate "real" enough representations.
- In HiAdv, the generator can be any HTC model with a text encoder and a structure encoder.
- It takes the global hierarchy as input and outputs a representation $\mathbf{h}_{\text{mix}}$ for multi-label classification.
- The generator is guided by the classification loss L_C.

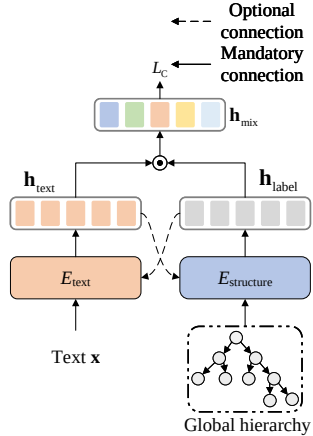

Figure 4: The generator of HiAdv.

HiAdv: Encoder

- The encoder generates an oracle representation with "real" input.
- In HiAdv, the encoder is trained in an auto-encoder manner: it encodes the local hierarchy along with text into the oracle representation and then reconstructs the local hierarchy.

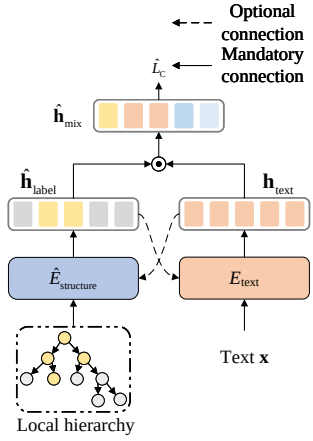

Figure 5: The encoder of HiAdv.

HiAdv: Encoder

- We modify the label embedding L of the structure encoder to contain the local hierarchy as a prior. For label i, we add an indicating embedding:

$$\hat{\mathbf{l}}_i = \mathbf{l}_i + \begin{cases} \mathbf{e}_1, & y_i \in Y \\ \mathbf{e}_0, & y_i \notin Y \end{cases} \tag{1}$$

- The oracle representation $\hat{\mathbf{h}}_{\text{mix}}$ is then reconstructed to the local hierarchy Y with a multi-label classifier.

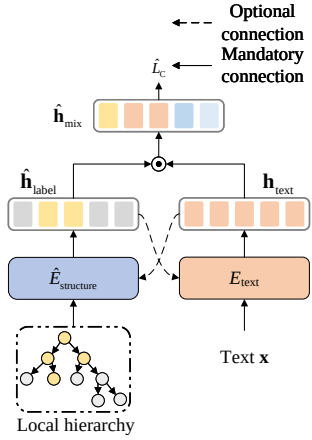

Figure 6: The encoder of HiAdv.

HiAdv: Discriminator

- The discriminator tries to distinguish the representation from the encoder $\hat{\mathbf{h}}_{\text{mix}}$ and the generator $\mathbf{h}_{\text{mix}}$.
- This is a binary classification task and the discriminator is guided by the binary cross-entropy loss:

$$L_{dis} = -(l \log p + (1 - l) \log(1 - p)) \tag{2}$$

where l is an indicator

$$l = \begin{cases} 0, & \mathbf{h} = \mathbf{h}_{\text{mix}} \\ 1, & \mathbf{h} = \hat{\mathbf{h}}_{\text{mix}} \end{cases} \tag{3}$$

and p is the predicted probability.

HiAdv: As a Whole

The generator, the encoder, and the discriminator train adversarially.

- The encoder and generator share the same text encoder and multi-label classifier.
- The discriminator aims to distinguish the representation while the generator attempts to "fool" the discriminator.
- Besides the classification loss, the generator is guided by an adversarial loss:

$$L_{adv} = -(1 - I) \log p \qquad (4)$$

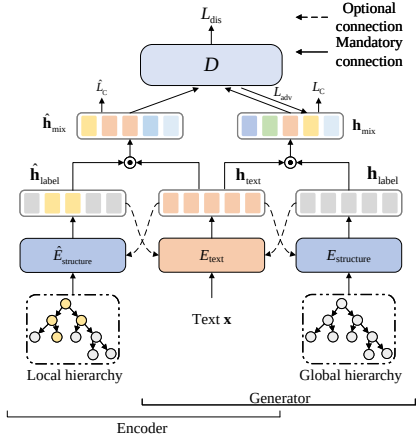

Figure 7: Training framework of HiAdv.

Experiments

We test on two backbone HTC models:

- HiBERT: BERT encoder followed by a graph encoder.
- HPT: One of the state-of-the-art models that adopts prompt tuning.

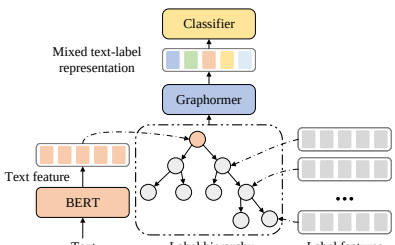

Figure 8: HiBERT

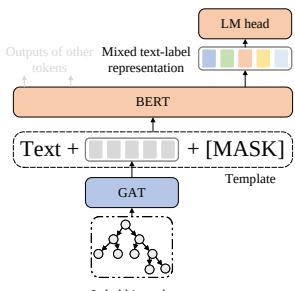

Figure 9: HPT

Main Results

Model	WOS (Depth 2)		RCV1-V2 (Depth 4)		NYT (Depth 8)	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
BERT [5]	85.63	79.07	85.65	67.02	78.24	66.08
BERT+HiAGM[5]	86.04	80.19	85.58	67.93	78.64	66.76
BERT+HTCInfoMax[5]	86.30	79.97	85.53	67.09	78.75	67.31
BERT+HiMatch [7]	86.70	81.06	86.33	68.66	-	-
HGCLR [5]	87.11	81.20	86.49	68.31	78.86	67.96
HPT [2]	87.16	81.93	87.26	69.53	80.42	70.42
HBGL [6]	**87.36**	**82.00**	87.23	**71.07**	80.47	70.19
HiBERT	85.77	80.10	86.49	68.82	79.49	68.40
HiBERT + HiAdv	86.38	80.78	86.74	69.43	79.56	69.30
HPT	87.08	81.59	86.96	69.25	80.21	70.14
HPT + HiAdv	87.20	81.62	**87.36**	69.62	**80.83**	**70.78**

Table 1: F1 scores on 3 datasets. The best results are in boldface.

Effect of Local Hierarchy

Our framework indeed learns from the local hierarchy.

- With it corrupted, results of both HiBERT and HPT drop significantly.
- Variants with more local hierarchy have better performance.
- HPT is a more powerful model, so it is more sensitive to the local hierarchy.

Variants	Micro-F1	Macro-F1
HiBERT + HiAdv	**80.00**	**70.68**
- w/ partial local hierarchy	79.51	69.64
- w/ no local hierarchy	78.46	68.09
- w/ wrong local hierarchy	78.67	67.02
HPT + HiAdv	**80.99**	**71.73**
- w/ partial local hierarchy	80.86	71.50
- w/ no local hierarchy	77.82	64.14
- w/ wrong local hierarchy	77.09	61.87

Table 2: Results of modifying local hierarchy on the development set of NYT dataset. The best results are in boldface.

Results on Imbalanced Hierarchy

We visualize Macro F1 scores of label clusters on the development set of NYT.

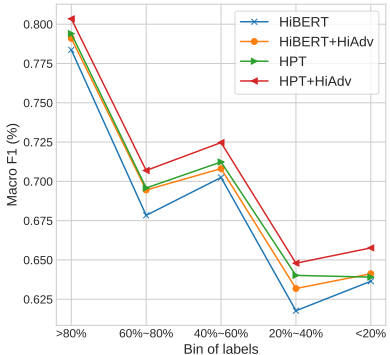

Figure 10: Label clusters grouped by the number of training samples.

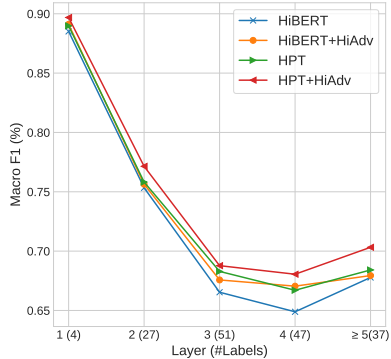

Figure 11: Label clusters grouped by depth in the hierarchy.

Conclusion

In this paper, we propose a hierarchy-aware adversarial framework (HiAdv) for assisting existing HTC models to incorporate local hierarchy.

- Our framework can be applied to any HTC model that involves a text encoder and a graph encoder.
- Experiments show that the adversarial framework is adept at dealing with complex hierarchies or promoting weak models that cannot fully learn the hierarchy.
- Further experiments demonstrate that the effect of HiAdv comes from the local hierarchy and the local hierarchy is beneficial for classes with deficient training instances.
- Our code is available at https://github.com/wzh9969/hiadv.

Acknowledgements

- We thank all the anonymous reviewers for their constructive feedback.
- This work was supported by the National Natural Science Foundation of China (62036001) and the National Science and Technology Major Project (2022ZD0116308).
- The slides are based on the templates from https://github.com/PM25/SimplePlus-BeamerTheme and https://www.overleaf.com/latex/templates/simpleplus-beamertheme/wfmfjhdcrdfx

Reference I

[1] J. Zhou, C. Ma, D. Long, *et al.*, "Hierarchy-aware global model for hierarchical text classification," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, Jul. 2020, pp. 1106–1117. DOI: 10.18653/v1/2020.acl-main.104. [Online]. Available: https://aclanthology.org/2020.acl-main.104.

[2] Z. Wang, P. Wang, T. Liu, *et al.*, "HPT: Hierarchy-aware prompt tuning for hierarchical text classification," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 3740–3751. [Online]. Available: https://aclanthology.org/2022.emnlp-main.246.

[3] Z. Deng, H. Peng, D. He, J. Li, and P. Yu, "HTCInfoMax: A global model for hierarchical text classification via information maximization," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online: Association for Computational Linguistics, Jun. 2021, pp. 3259–3265. DOI: 10.18653/v1/2021.naacl-main.260. [Online]. Available: https://aclanthology.org/2021.naacl-main.260.

Reference II

[4] R. Zhao, X. Wei, C. Ding, and Y. Chen, "Hierarchical multi-label text classification: Self-adaption semantic awareness network integrating text topic and label level information," in *International Conference on Knowledge Science, Engineering and Management*, Springer, 2021, pp. 406–418. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-82147-0_33.

[5] Z. Wang, P. Wang, L. Huang, X. Sun, and H. Wang, "Incorporating hierarchy into text encoder: A contrastive learning approach for hierarchical text classification," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 7109–7119. DOI: 10.18653/v1/2022.acl-long.491. [Online]. Available: https://aclanthology.org/2022.acl-long.491.

[6] T. Jiang, D. Wang, L. Sun, Z. Chen, F. Zhuang, and Q. Yang, "Exploiting global and local hierarchies for hierarchical text classification," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 4030–4039. [Online]. Available: https://aclanthology.org/2022.emnlp-main.268.

Reference III

[7] H. Chen, Q. Ma, Z. Lin, and J. Yan, "Hierarchy-aware label semantics matching network for hierarchical text classification," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online: Association for Computational Linguistics, Aug. 2021, pp. 4370–4379. DOI: 10.18653/v1/2021.acl-long.337. [Online]. Available: https://aclanthology.org/2021.acl-long.337.