Evaluating Hierarchical Document Categorisation

Qian Sun[♣] **Aili Shen**[♣] **Hiyori Yoshikawa**[♢] **Chunpeng Ma**[♢]
Daniel Beck[♣] **Tomoya Iwakura**[♢] **Timothy Baldwin**[♣]
[♣] The University of Melbourne
[♢] Fujitsu Limited
qiasun@student.unimelb.edu.au, {aili.shen, d.beck, tbaldwin}@unimelb.edu.au
{y.hiyori, ma.chunpeng, iwakura.tomoya}@fujitsu.com

Abstract

Hierarchical document categorisation is a special case of multi-label document categorisation, where there is a taxonomic hierarchy among the labels. While various approaches have been proposed for hierarchical document categorisation, there is no standard benchmark dataset, resulting in different methods being evaluated independently and there being no empirical consensus on what methods perform best. In this work, we examine different combinations of neural text encoders and hierarchical methods in an end-to-end framework, and evaluate over three datasets. We find that the performance of hierarchical document categorisation is determined not only by how the hierarchical information is modelled, but also the structure of the label hierarchy and class distribution.

1 Introduction

Document categorisation is a core task in information retrieval and natural language processing, whereby documents are categorised relative to a pre-defined set of labels. While the majority of research on document categorisation assumes a flat label structure, in practice in large-scale document categorisation tasks, there is often hierarchical label structure, in the form of either a tree or directed acyclic graph (Zhou et al., 2020; Azarbonyad et al., 2021), where "child" labels inherit the properties of their parents. The goal of hierarchical document categorisation is to classify documents into a set of labels, where there is a hierarchical relationship among the labels.

Hierarchical document categorisation methods explicitly capture the label structure during training. There has been a resurgence of interest in document categorisation in recent years, in part driven by breakthroughs in representation learning and pre-trained language models (Mikolov et al., 2013; Pennington et al., 2014; Peters et al., 2018; Kim, 2014; Wang et al., 2017; Devlin et al., 2019), which generate more expressive, general-purpose representations, thereby leading to performance gains across a range of NLP tasks. Despite this, there has been relatively little recent work specifically on hierarchical document categorisation. What recent work does has varied wildly in the choice of text encoder and dataset, with no systematic, controlled cross-dataset evaluation to be able to make solid conclusions as to whether the reported performance gains are attributable to the proposed hierarchical document categorisation method or just the text encoders used. Our work focuses on examining the capacity of existing methods dealing with labels with a hierarchical structure, which is different from the work of Yang et al. (2016), which focuses on modelling documents in a hierarchical way to perform classic document classification task.

In this work, we carry out systematic evaluation of a range of contemporary hierarchical document categorisation approaches, using a range of neural text encoders, based on three document collections with hierarchical label sets.

2 Related Work

Hierarchical document categorisation methods can be grouped into: flat approaches, local approaches, global approaches, and hybrid methods, based on how they utilise the label hierarchy.

2.1 Flat Approaches

Flat approaches (Eisner et al., 2005; Freitas and Carvalho, 2007) simply ignore the label hierarchy, and assume all classes are independent. As such, they are unable to capture the label structure and are poor at handling mutual exclusivity, especially among sibling nodes in multi-label categorisation tasks.

2.2 Local Approaches

Local approaches generally make predictions top-down recursively, along paths in the label hierarchy. They can be divided into three groups (Silla and Freitas, 2011): a local classifier per node (LCN), a local classifier per parent node (LCPN), or a local classifier per level (LCL). In LCN, there is a binary classifier for each node, which determines whether a document belongs to that node or not (Eisner et al., 2005; Freitas and Carvalho, 2007). In contrast, LCPN (Davies et al., 2007; Secker et al., 2010; Shimura et al., 2018; Banerjee et al., 2019) employs a multi-class classifier at each parent node, predicting which child node the document should be assigned to. Compared with LCN, LCPN significantly reduces the number of local classifiers, and can be applied in either single-label or multi-label settings. In contrast, LCL (Kowsari et al., 2017) employs a multi-class classifier at each layer in the hierarchy. This method usually fails to capture parent–child information between layers. For all three approaches, a top-down approach is often used to avoid label inconsistency, making them prone to error propagation.

2.3 Global Approaches

Global approaches (Mao et al., 2019; Zhou et al., 2020) optimise across all labels simultaneously, taking the label hierarchy into account. The simplest global approach converts the hierarchical categorisation task into a multi-label categorisation task, where each original label is replaced with its ancestors and itself. Similar to local approaches, this potentially results in label inconsistency during inference. A more popular global approach is to include a loss term which captures the hierarchy in some way (Gopal and Yang, 2013; Peng et al., 2018), such as an entropy term (Clare and King, 2003) or distance metric (Vens et al., 2008). For example, Zhou et al. (2020) proposed a hierarchy-aware structure encoder to model the label hierarchy as a directed graph. It can capture global hierarchical information as it models both top-down and bottom-up label dependencies. Moreover, all nodes are linked with each other, meaning that pairwise co-occurrence can be modelled in addition to parent–child relationships.

2.4 Hybrid Methods

There are also hybrid methods which combine the methods mentioned above (Wehrmann et al., 2018; Huang et al., 2019). For example, Gopal and Yang (2013) used simple recursive regularisation to encourage parameter smoothness between linked nodes, with positive results independently reported by Peng et al. (2018) and Zhou et al. (2020).

3 Experiments

3.1 Models

In our work, each model consists of a text encoder and a hierarchical method, where the text encoder is used to obtain text representations, and the hierarchical method makes predictions with the assistance of hierarchical label information.

3.1.1 Text Encoders

TextCNN (Kim, 2014): A CNN made up of convolutional and max-pooling layers. In this work, we apply convolution kernels with width 2, 3, and 4 (3 for each width size) to word embeddings, and use a max-pooling layer.

TextRNN: A single-layer Bi-LSTM (Wang et al., 2017) with a cell size of 64 where the concatenated hidden state at the last timestep makes up the document representation.

TextRCNN: A combination of TextCNN and TextRNN, where we first employ a single-layer Bi-LSTM with a cell size of 64 and obtain outputs across all timesteps by concatenating outputs from both directions, then apply convolution kernels with width 2, 3, and 4 (3 for each width size), followed by a max-pooling layer. This method has achieved state-of-the-art on RCV1 for both flat and hierarchical categorisation (Zhou et al., 2020).

BERT (Devlin et al., 2019): The hidden state of "CLS" from BERT is used as the document representation, using the base-uncased version.

3.1.2 Hierarchical Methods

Flat: Baseline method where all nodes are treated as candidate classes, ignoring hierarchical information.

Recursive Regularization (RR: Gopal and Yang (2013)): A hybrid method, utilising simple recursive regularisation to encourage parameter smoothness between linked nodes.

Hierarchical Multi-Label Classification Networks (HMCN: Wehrmann et al. (2018)): A hybrid local/global approach, where each level in

Dataset	|L|	Avg(|L|)	Depth	Training	Test
RCV1	103	3.24	4	23,149	592,688
SHINRA	237	3.16	4	390,433	43,382
WoS	141	2.00	2	42,286	4,699

Table 1: Statistics of datasets: "|L|" is the total number of labels; "Avg(|L|)" is the average number of labels per document; and "Depth" indicates the maximum hierarchy depth.

the model corresponds to a level in the label hierarchy. The global model consists of multiple linear layers with ReLU as the activation function. The input to each layer includes the original sequence and the output from its immediate last layer, where the hidden size for each layer is 384 as in Wehrmann et al. (2018). Passing information from the first layer to the last layer, we obtain the global output. In addition, the output from each layer is also fed into a local layer, where the hidden size is the number of nodes/classes in the corresponding hierarchical level. Then the sum of the global output and concatenated local outputs is fed into a sigmoid function to predict the classes.[1]

Hi-GCN (Zhou et al., 2020): An end-to-end hierarchy-aware global model that extracts the label hierarchy information to achieve label-wise text features. A graph convolutional network is used as the structure/hierarchy encoder, where each edge represents the correlation between a pair of nodes. There are three types of edges in the graph: top-down, bottom-up, and self-loop edges, where the weights for bottom-up and self-loop edges are 1, and the weights for top-down edges are determined by the predefined hierarchy and dataset distributions. To obtain label-wise text features, hierarchical text feature propagation is used. Specifically, the text representation from a text encoder is reshaped to act as the node input, which is updated through the hierarchy-aware structure encoder. The output of a node is based on its neighbourhood: itself, its child nodes, and its parent nodes. The output hidden state is then fed into the final classifier.

[1]In the original work of Wehrmann et al. (2018), the authors first apply the sigmoid function to the global output and local outputs, respectively, resulting into extremely bad performance in some settings, indicating that applying sigmoid separately to the global and local outputs is not as effective as applying it to the combined global and local information.

3.2 Datasets

We evaluate each text encoder+hierarchical method combination in an end-to-end framework over three datasets: RCV1 (Lewis et al., 2004), SHINRA (Sekine et al., 2020), and WoS (Kowsari et al., 2017). Here, RCV1 is a collection of news articles published by the Reuters News between 1996 and 1997. SHINRA contains English Wikipedia articles from the SHINRA2020-ML shared-task (Sekine et al., 2020), where each Wikipedia article is labelled according to a fine-grained named entity label set known as Extended Named Entity (ENE).[2] WoS is a collection of abstracts from academic papers across different research domains and areas. The statistics of each dataset is given in Table 1. Looking at the document distributions in terms of label hierarchy levels, we find that the relationship between the number of documents and label classes conforms to a power-law function for RCV1 and SHINRA, especially at lower (2+) levels. For WoS, the number of documents per class at level 1 and 2 is relatively balanced.

3.3 Evaluation Metrics

We evaluate model performance in terms of Micro-F_1 and Macro-F_1, two standard evaluation metrics for document categorisation. Micro-F_1 is instance-level F-score, and thereby gives more weight to frequent labels. Macro-F_1 is class-level F-score, and gives equal weight to all labels.

3.4 Experimental Settings

Each document is truncated/padded to a fixed length of 256 tokens, where stopwords are removed for all models except BERT. For all models except BERT, we use 100-dimensional pre-trained word embeddings from GloVe (Pennington et al., 2014) to initialise the word embeddings. The vocabulary contains at most 100,000 words ranked by frequency. For OOV words, the word embeddings are randomly initialised. We train all models with

[2]http://ene-project.info/ene8/?lang=en

Dataset	RCV1		SHINRA		WoS	
Method	Micro	Macro	Micro	Macro	Micro	Macro
TextCNN						
Flat	75.63	45.24	86.94	56.46	83.41	77.00
RR	75.56	50.81	85.31	56.62	83.51	77.32
HMCN	**78.22**	43.49	**87.03**	56.28	80.24	74.38
Hi-GCN	77.80	**51.34**	86.91	**58.61**	**84.09**	**77.37**
TextRNN						
Flat	78.46	49.18	88.43	60.11	83.72	77.55
RR	78.52	55.48	87.22	60.07	83.57	**78.08**
HMCN	80.52	48.97	88.71	59.76	82.09	75.90
Hi-GCN	**81.57**	**56.29**	**88.74**	**61.20**	**84.11**	77.95
TextRCNN						
Flat	79.92	51.54	88.12	60.34	84.05	77.95
RR	79.81	56.37	88.06	60.32	84.14	78.03
HMCN	81.13	50.44	88.56	59.71	82.86	76.11
Hi-GCN	**82.96**	**58.05**	**88.69**	**61.05**	**84.54**	**78.28**
BERT						
Flat	82.64	55.61	90.86	66.35	75.73	69.22
RR	82.13	59.41	90.70	66.59	75.77	69.43
HMCN	82.68	53.65	91.32	64.13	72.28	64.62
Hi-GCN	**83.20**	**60.32**	**91.90**	**67.79**	**75.94**	**70.81**

Table 2: Experimental results for different combinations of encoders and hierarchical document categorisation methods. The best result for each text encoder on each dataset is indicated in bold. Micro and Macro indicate micro and macro F_1 score, resp..

a batch size of 32 using Adam (Kingma and Ba, 2014), and an initial learning rate of 1e-3 (1e-5 for BERT) for at most 20 epochs.

For hierarchical categorisation methods, the penalty coefficient of recursive regularisation is set to 1e-6, while the output dimension of internal linear layers in HMCN is set to 384. For the hyper-parameters of Hi-GCN, we follow the recommendations of the authors in the original paper (Zhou et al., 2020). Note that in some cases, both HMCN and Hi-GCN suffer from the vanishing/exploding gradient problem, to counter which we apply batch normalisation to the outputs of the linear layers in HMCN and Hi-GCN where necessary.

3.5 Results

Table 2 presents the experimental results of different combinations of text encoders and hierarchical categorisation methods across the three datasets. Model performance is heavily influenced by the choice of text encoder, with BERT outperforming other encoders by a large margin on RCV1 and SHINRA in terms of both Micro-F_1 and Macro-F_1, but underperforming on WoS, irrespective of which hierarchical method it is combined with. We hypothesis that the performance drop for BERT on WoS is mainly due to domain shift, in that it has been pre-trained on Wikipedia articles and the Google Books corpus, which differ substantially from academic writing.[3] Among TextCNN, TextRNN, and TextRCNN, TextCNN underperforms TextRNN and TextRCNN on all three datasets, especially on RCV1 and SHINRA. The reason is that TextCNN can only capture local features, but the fine-grained hierarchical distinctions captured in the different label sets often require longer-distance semantic dependencies.

With regards to the hierarchical categorisation methods, compared with Flat on RCV1 and

[3] It would be interesting to experiment with SciBERT (Beltagy et al., 2019), which has been pre-trained on papers from the scientific domain, which we leave to future work.

SHINRA, RR improves Macro-F_1 in most cases at the cost of Micro-F_1, indicating that RR can improve the performance of classes with fewer training samples. In contrast, HMCN improves Micro-F_1 at the cost of Macro-F_1, indicating that HMCN is biased towards classes that are better represented in the dataset. However, on WoS, RR achieves better performance in terms of both Micro-F_1 and Macro-F_1— with the one exception of Micro-F_1 with TextRNN— while HMCN achieves worse performance in terms of both Micro-F_1 and Macro-F_1. All these results can be attributed to the fact that RR and HMCN leverage hierarchical information differently: RR utilises parent–child relationships, while HMCN adopts layer-wise hierarchical information. As a result of error propagation due to the greedy top-down approach, HMCN performs relatively worse the deeper the label hierarchy. For example, Flat with TextCNN achieves a Micro-F_1 of 88.53 at level-1 (7 classes) and a Micro-F_1 of 83.41 at level-2 (134 classes) on WoS, where both Micro-F_1 scores at these two levels are higher than 80.24 achieved by HMCN, indicating that the categorisation errors of HMCN at level-1 propagate to level-2 and lead to worse results on WoS.

Looking to Hi-GCN, we find that Hi-GCN with any text encoder consistently outperforms other methods on all three datasets in terms of both Micro-F_1 and Macro-F_1, by aggregating hierarchical information in a more flexible way. In addition to passing information from parent to child nodes, it also passes information from child to parent nodes, thereby improving categorisation performance at level-1 and categorisation at subsequent levels. Both RCV1 and SHINRA datasets have extremely imbalanced data distributions while WoS is relatively more balanced, which is also revealed by the greater differences between Micro-F_1 and Macro-F_1 on RCV1 and SHINRA, than on WoS.

These experiments indicate that the performance of hierarchical document categorisation not only depends on the text encoder and particular hierarchical methods, but also the intrinsic hierarchy label structure and the label distribution.

4 Conclusions

We examine various combinations of text encoders and hierarchical categorisation methods in an end-to-end fashion over three datasets. We find that the choice of text encoder is a strong determinant of performance than the choice of hierarchical method, and indeed that local hierarchical methods don't consistently outperform baseline flat classification methods. With regards to hierarchical methods, RR improves Macro-F_1 at the cost of Micro-F_1 on RCV1 and SHINRA, while HMCN improves Micro-F_1 at the cost of Macro-F_1 on RCV1 and SHINRA. An opposite trend is observed on WoS, namely an improvement for RR and deterioration for HMCN. These different behaviours are determined by how the hierarchical label information is modelled during training. The global model Hi-GCN achieves superior performance in terms of both Micro-F_1 and Macro-F_1 on all three datasets, indicating the necessity of capturing the hierarchy label structure holistically.

References

Hosein Azarbonyad, Mostafa Dehghani, Maarten Marx, and Jaap Kamps. 2021. Learning to rank for multi-label text classification: Combining different sources of information. *Natural Language Engineering*, 27(1):89–111.

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.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3615–3620.

Amanda Clare and Ross D King. 2003. Predicting gene function in saccharomyces cerevisiae. *Bioinformatics*, 19(suppl_2):ii42–ii49.

Matthew N Davies, Andrew Secker, Alex A Freitas, Miguel Mendao, Jon Timmis, and Darren R Flower. 2007. On the hierarchical classification of G protein-coupled receptors. *Bioinformatics*, 23(23):3113–3118.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Roman Eisner, Brett Poulin, Duane Szafron, Paul Lu, and Russell Greiner. 2005. Improving protein function prediction using the hierarchical structure of the gene ontology. In *Proceedings of the 2005 IEEE*

Symposium on Computational Intelligence in Bioinformatics and Computational Biology, pages 1–10.

Alex Freitas and André Carvalho. 2007. A tutorial on hierarchical classification with applications in bioinformatics. *Research and Trends in Data Mining Technologies and Applications*, pages 175–208.

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.

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.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

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 *16th IEEE International Conference on Machine Learning and Applications*, pages 364–371.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5(Apr):361–397.

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*, pages 445–455.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in Neural Information Processing Systems 26*, pages 3111–3119.

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.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Andrew Secker, Matthew N Davies, Alex Alves Freitas, EB Clark, Jonathan Timmis, and Darren R Flower. 2010. Hierarchical classification of G-protein-coupled receptors with data-driven selection of attributes and classifiers. *International Journal of Data Mining and Bioinformatics*, 4(2):191–210.

Satoshi Sekine, Masako Nomoto, Kouta Nakayama, Asuka Sumida, Koji Matsuda, and Maya Ando. 2020. Overview of shinra2020-ml task. In *Proceedings of the NTCIR-15 Conference*.

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):31–72.

Celine Vens, Jan Struyf, Leander Schietgat, Sašo Džeroski, and Hendrik Blockeel. 2008. Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2):185.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 4144–4150.

Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. 2018. Hierarchical multi-label classification networks. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5075–5084.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

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.