A Study on Hierarchical Text Classification as a Seq2seq Task

Fatos Torba, Christophe Gravier, Charlotte Laclau, Abderrhammen Kammoun, Julien Subercaze

A Study on Hierarchical Text Classification as a Seq2seq Task

Fatos Torba[1,2][0009−0002−6947−0855], Christophe Gravier[2][0000−0001−8586−6302], Charlotte Laclau[3][0000−0002−7389−3191], Abderrhammen Kammoun[1], and Julien Subercaze[1]

[1] AItenders
{fatos.torba, abderrhammen.kammoun, julien.subercaze}@aitenders.com
[2] Laboratoire Hubert Curien, UMRCNRS 5516 Saint-Etienne, France
christophe.gravier@univ-st-etienne.fr
[3] Télécom Paris, Institut Polytechnique de Paris, Paris, France
charlotte.laclau@telecom-paris.fr

Abstract. With the progress of generative neural models, Hierarchical Text Classification (HTC) can be cast as a generative task. In this case, given an input text, the model generates the sequence of predicted class labels taken from a label tree of arbitrary width and depth. Treating HTC as a generative task introduces multiple modeling choices. These choices vary from choosing the order for visiting the class tree and therefore defining the order of generating tokens, choosing either to constrain the decoding to labels that respect the previous level predictions, up to choosing the pre-trained Language Model itself. Each HTC model therefore differs from the others from an architectural standpoint, but also from the modeling choices that were made. Prior contributions lack transparent modeling choices and open implementations, hindering the assessment of whether model performance stems from architectural or modeling decisions. For these reasons, we propose with this paper an analysis of the impact of different modeling choices along with common model errors and successes for this task. This analysis is based on an open framework coming along this paper that can facilitate the development of future contributions in the field by providing datasets, metrics, error analysis toolkit and the capability to readily test various modeling choices for one given model.

Keywords: Hierarchical text classification · generative model · reproducibility.

1 Introduction

Hierarchical Text classification (HTC) aims at tagging an input text with labels organized in an external label tree. It differs from standard text classification as an arbitrary number of labels are expected in the predictions, and also because the predictions have to be consistent with respect to the label tree [13]. HTC, which has long been challenging for machine learning and natural language processing, is also deemed mandatory in various industrial settings such as e-commerce applications [4], medical recording [2], and finance [8], to name a few.

Despite the advent of Large Language Models (LLMs), HTC is still considered a difficult task [1]. One initial reason is that this task involves classifying items into an arbitrary number of categories. Furthermore, this task is akin to choosing one or multiple, potentially incomplete, pathways from a class tree. Consequently, the hierarchical classifier must exhibit the ability to classify at a coarse-grain as well as a fine-grained level, which requires attention capacities proportional to the depth of the class tree. Moreover, the training data are imbalanced since the annotations for the leaves

in the class tree are inevitably sparse due to the hierarchical structure [7], but also since the annotations can stop at any pre-leaf level by tagging design. Finally, there exists very few large-scale open datasets for hierarchical classification with respect to datasets available in standard text classification, due to the cost of such complex annotations and the straightforward industrial applications that can directly result from it. From this complexity, two main opportunities also arise to solve this task. The first one consists in leveraging the hierarchical information between classes to guide the model towards consistent and easier decisions [18,9,21,20]. The second subsists in exploiting the possible semantic cues between label tokens in the class tree path to predict and the text to classify, which are even sometimes expanded with label metadata [16,5,3,15].

Throughout the years, a diverse set of approaches and methodologies have been devised by researchers to address the challenge of HTC. These contributions are traditionally divided between global [3,18,9] and local approaches [17]. Local approaches consist of learning a set of localized classifiers that are learned for each node, for each parent node, or for each class tree level. The major shortcoming is the number of classifiers to learn as the class tree grows and the difficulty to accurately predict leaf nodes, which suffer from sparse annotations although they represent a fine-grain classification to perform. Contrary to local approaches, global approaches use one single model to capture the hierarchical information [21]. These approaches either flatten the class hierarchy as a single array of classes to predict or classify one node at a time, possibly constraining the next prediction to the ones already available for the previously predicted levels [6].

For this latter case, since the aim is to predict one or several paths from the class tree, the prediction can be seen as a sequence of label tokens, thus allowing to cast the problem as a sequence-to-sequence (seq2seq) task [15,1,19,20]. This is the setting that we are investigating in this paper because of the progress of seq2seq LLMs applied to HTC.

In what follows, SHTC will refer to "seq2seq HTC" models. Besides neural architectural choices, original and specific to each SHTC classifier, there exists a shared set of modeling choices to make when modeling the problem as a seq2seq task. While the literature lacks an in-depth exploration of the impact of these modeling choices, our paper addresses this gap by conducting a study on the three following key modeling choices for SHTC.

Modeling choice #1: First, there are different ways to traverse the nodes in the tree when producing the gold standard of sequences: 1) horizontally (BFS algorithm) or vertically (DFS algorithm), and 2) applying these algorithms from the root to the leaves as it is usually performed, but one can also consider traversing from the leaves to the root so that the hardest tokens (leaves are fine-grain classification levels) are at the beginning of the beam search at decoding time, trying to enforce more attention to them when training. The cartesian product of these two considerations for class tree traversal leads to four different options as illustrated in Figure 1 ("Target Generator").
Modeling choice #2: SHTC can also lead to hierarchy incoherent prediction. One can choose to constrain or free the next token prediction to be based on the current predictions to enforce hierarchy consistency.
Modeling choice #3: Additionally the label name can encapsulate a semantic similarity with the input text. The model may easily predict the name of the labels rather than for instance their assigned acronym. The impact of such a modeling choice (name versus acronym) is yet unknown.

In this paper, we propose an open framework that allows testing the different variations for these three modeling choices and that we hope can facilitate the development of future contributions in the field by providing datasets, metrics, an error analysis toolkit, and the capability to readily test various modeling choices for one given model (Sec. 2). We also propose an evaluation of how the

three main modeling choices for SHTC impact their performance (Sec. 3). The paper comes with an open source framework[4] which makes it possible to plug one's HTC model and run their own experiments on the provided datasets and perform error analysis, for all possible options of these three modeling choices.

2 Analysis Framework

Problem Definition Given an input text $X = \{x_1, x_2, .., x_n\}$ and a taxonomic hierarchy $T = \{V, E\}$ where n, V and E denote respectively the text length, the labels nodes set and the parent-child relationship, SHTC aims to predict $Y = \{y_1, y_2, \ldots, y_k\}$ where k is the number of true labels, y_i are the target nodes of X and Y a subset of V. We can model this problem as a seq2seq task by predicting the Y as a sequence conditioned on the given input token sequence X, so that $p(y_1, y_2, \ldots, y_k|X) = \prod_{t=1}^{k} p(y_t|y_{<t}, X)$, which is in turn approximated via beam search in practice.

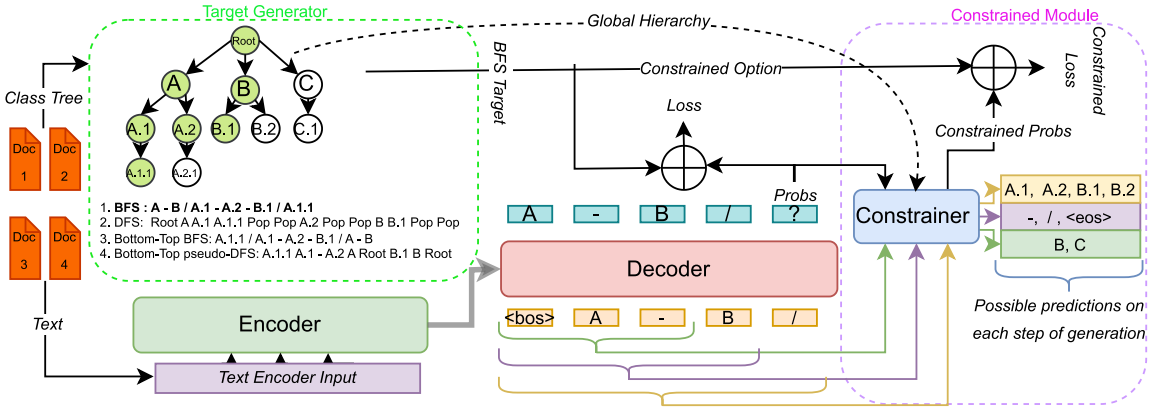

Fig. 1. SHTC illustration. The Target generator offers 4 target sequence options, green nodes denote correct document labels. The constrained module is optional, without it, SHTC operates like a standard T5 architecture. During auto-regressive decoding, this latter computes the probability distribution over the next possible tokens set based on prior predictions and global hierarchy, while nullifying others (constrained probs). To enhance clarity, we only consider a BFS target. For instance, if the model has already generated "A -" the constrained module will distribute the probabilities over {B, C} (indicated by the green arrow and green box). The purple and yellow arrows illustrate the next 2 constrained generation steps.

2.1 Sequence Modeling

Since the target sequence can be constructed in various ways we investigate in this paper four options (Fig. 1 "Target Generator"). The Depth-First Search (DFS) starts at the root and explores as far as possible along each branch before backtracking ("Pop"). Breadth-First Search (BFS) explores all nodes at the present level prior to moving on to the next one. The intra-level relationship is modeled here by "-", while "/" signifies an inter-level relationship. Bottom-Top BFS is the same process

[4] `https://github.com/FatosTorba/SHTC`

as BFS but starts from the deepest level. Bottom-Top pseudo-DFS starts from the left deepest node and moves either horizontally if it has a brother node (except for the "Root" parent) or upper if not and so on until all true nodes have been explored. Here "-" represents brother relationship.

Following the state of art [7,20] we choose T5 [14], a pre-trained Transformer-based architecture, as the backbone model. The presence of four target options results in the training of four distinct T5 models, each of which is given a specific name: BFS, BT-BFS, DFS, and BT-DFS, corresponding to BFS, Bottom-Top BFS, DFS, and Bottom-Top pseudo-DFS target sequences, respectively. The decoder can be constrained during both training and inference phases to generate a dynamic vocabulary as in [20]. We will provide a detailed description of this constrained decoding strategy in the following section.

2.2 Decoding Strategies

The decoder predicts a sequence of text tokens in an auto-regressive manner, one token at each step, starting from $<bos>$ (begin of sentence) token and ending when the $<eos>$ (end of sentence) token is predicted. At the i^{th} generation step of a token y_i one can decide to constrain the latter to belong to a given set of tokens.

Let $T = \{V, E\}$ be the class taxonomic hierarchy and S a set of special tokens used to build the target sequence. Let $y_0, y_1, ..., y_{i-1} = Y_{i-1} \in V \cup S$ be the set of previous generated tokens. Suppose we can construct a function $f_{constrain}(Y_{i-1}, T) = C_T \in V \cup S$ where C_T is the possible next tokens set. Constraining y_i to belong to C_T is equivalent to computing the probability distribution over C_T tokens using $Softmax$ function as follow: $\forall k \in \{1, n\}, \quad p(y_i = t_k | Y_{i-1}, T) = \frac{exp(l_k)}{\sum_{t_j \in C_T} exp(l_j)}$, if $t_k \in C_T, 0$ otherwise, where n, t_k and l_k are respectively the entire vocabulary length, the k^{th} token and k^{th} logit value of the decoder output. Thus, during training, the model will focus more towards discriminating the possible tokens rather than learning not to predict those not included in C_T. It reduces the optimization space since only the C_T tokens will influence the $Loss$ function. During inference, it ensures a coherent prediction with respect to the hierarchy. The $f_{constrain}$ function depends on the choice of the target sequence. For the BFS target Fig.1 shows an example of the output set of $f_{constrain}$ at the right of "Constrainer Module". For the DFS target $f_{constrain}$ will return the child's tokens set of the previous predicted parent or the backtracking token. For the "Root" node the $<eos>$ is also a possible token to stop the generation where no more nodes of level one can be predicted.

2.3 Proposed metrics

Inspired by existing works [20], we use four sets of metrics for SHTC evaluation:
F1 scores: F1-micro, F1-macro, per level and global.
C-F1 scores: (C-F1 micro, C-F1 macro, per level and global) corresponding respectively to F1-micro and F1-macro of constrained predictions obtained by post-processing, considering a node as `true` only if all its ancestor nodes have been predicted as `true`.
Hierarchical inconsistencies: Hierarchical Consistency Rate (HCR), True Positive Hierarchical Consistency Rate (TP-HCR) and False positive Hierarchical Consistency Rate (FP-HCR). HCR refers to the proportion of predicted labels that align with the parent-child relationships of labels within a predefined hierarchy. TP-HCR and FP-HCR refer to HCR of predicted labels where we

remove respectively the false positive predicted labels and true positive predicted labels.

Depth of Prediction Rate: computed as DPR= $\frac{1}{|D|}\Sigma_{i=1}^{|D|} depth(y_{pred,i}) \geq depth(y_{true,i})$ where D is a set of documents, $y_{pred,i}$ and $y_{true,i}$ are respectively the predicted labels and the true labels of document i and *depth* is a function that gives the deepest level number of a predicted label. A higher DPR suggests that the model tries to classify fine-grained labels and conversely a smaller one indicates that the model tends to focus more on higher-level categories and avoid taking the risk of predicting low-level label predictions.

3 Experiment and Datasets

3.1 Datasets and implementation details

Datasets We conduct experiments on three public datasets, including reuters corpus (RCV1V2 [5] [11]), Blurb Genre Collection (BGC[6]) and Web-Of-Science (WOS[7] [10]). The labels of all this dataset are organized as a tree-like hierarchy. RCV1V2 and BGC are both multi-paths while WOS is single-path. More details for the datasets are shown in Table 1.

Dataset	$\|V\|$	*Depth*	$m\|V\|$	mD	mW_{max}	$mW(L_1/L_2/L3/L_4)$	$n(L_2/L_3/L_4)$	Train	Val	Test
RCV1V2	103	4	3.24	2.56	1.45	1.18/1.45/1.23/1	779702/452238/23211	20833	2316	781265
WOS	141	2	2	2	1	1/1/-/-	46798/0/0	30070	7518	9397
BGC	146	4	3.01	2.34	1.51	1.06/1.46/1.39/1.08	77438/45005/3213	58715	14785	18394

Table 1. Statistics of datasets: L_j is the level j, $|V|$ the total number of labels, *Depth* is the maximum level, $m|V|$, mD, mW_{max}, $mW(L_1/L_2/L_3/L_4)$ are the respective averages of the number, depth, maximum width and width of L_1, L_2, L_3, and L_4 of labels in each sample. $n(L_2/L_3/L_4)$ stands for the number of samples in the overall dataset for L_2, L_3, and L_4 (value for L_1 is the sum of train, val and test).

Implementation details For all datasets, we use the text description as input, ignoring all other metadata such as the author, title, or keywords for instance. Label names are kept pristine (no pre-processing). We trained T5 during 12 epochs with a batch size of 16 on a single GPU with a learning rate of $5e^{-5}$ and optimizing with AdamW [12]. During inference, we use the beam search strategy for sequence generation. For BGC we used the official split and partitioned the RCV1V2 training dataset into training and validation sets. For WOS, we made a random split into training, validation and test sets as no official split exists. We applied both constrained and not-constrained decoding strategies for BFS and DFS target sequences. For Bottom-Top BFS and Bottom-Top pseudo-DFS, no constrained decoding strategy is used since we want the model to learn the hierarchy. Constraining would reduce the possible next tokens set to parent tokens only. For RCV1V2 we conduct experiments using nodes designated by acronyms (only for the top two models, DFS and DFS-C), as presented by default in the dataset, as well as their corresponding full names in a dedicated experiment to assess the impact of semantic similarity between the input text and the label names.

[5] `http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm`
[6] `https://www.inf.uni-hamburg.de/en/inst/ab/lt/resources/data/blurb-genre-collection.html`
[7] `https://data.mendeley.com/datasets/9rw3vkcfy4/2`

3.2 Results

Model	WOS					RCV1V2					BGC				
	Mi. F1	Ma. F1	C-Mi. F1	C-Ma. F1	DPR	Mi. F1	Ma. F1	C-Mi. F1	C-Ma. F1	DPR	Mi. F1	Ma. F1	C-Mi. F1	C-Ma. F1	DPR
BFS	86.12	80.69	**86.12**	**80.69**	99.89	85.59	64.92	**85.59**	64.92	**97.00**	75.76	59.63	75.76	59.63	92.92
BFS-C	86.61	80.96	-	-	**99.97**	85.47	62.20	-	-	94.15	76.84	60.89	-	-	88.36
BT-BFS	85.62	80.34	85.61	80.32	99.79	84.51	64.74	84.52	64.74	96.35	75.67	59.47	75.67	59.48	**93.53**
DFS	85.90	80.20	85.91	80.21	99.93	85.51	65.59	85.52	**65.59**	96.75	76.54	60.37	**76.54**	**60.37**	93.14
DFS-C	**86.42**	**81.67**	-	-	99.93	**85.61**	**68.01**	-	-	96.75	**77.79**	**62.41**	-	-	92.89
BT-DFS	85.87	80.56	85.88	80.57	99.70	84.91	64.91	84.92	64.91	95.99	75.59	58.68	75.59	58.68	93.31
DFS-Acr	-	-	-	-	-	85.48	61.90	85.48	60.91	95.26	-	-	-	-	-
DFS-C-Acr	-	-	-	-	-	85.56	62.54	-	-	96.37	-	-	-	-	-

Table 2. Global results, "-Acr" stands for models predicting the acronym nodes of RCV1V2.

Experimental results are elaborated in Table 2 and Table 3. The constrained decoding strategy consistently improves the performance at each level and globally while assuring, by definition, coherent prediction. The sole exception is RCV1V2, where BFS-C performs worse than BFS: as indicated by the DPR and F1 scores at Level 3, BFS-C terminates generation at higher levels. All sequence modeling choices encapsulate the hierarchical structure of the taxonomy, making it learnable for T5 and consequently, the C-F1 scores are almost equal to F1 ones. DFS modeling performs better but the fact that the class sub-trees have bigger average depth than average width makes it unclear whether this improvement stems from the sequence modeling or the dataset properties. Bottom-top approaches decrease performance despite improvement in the last level, hierarchical coherency and depth rate. It is worth mentioning that the label names semantic information is crucial for the decoder. Predicting the RCV1V2 acronym taxonomy, as opposed to the full label names, significantly lowers the Macro F1 by approximately 4% for DFS and 5% for DFS-C.

Model	Mi. F1	Ma. F1	C-Mi. F1	C-Ma. F1	TP-HCR	FP-HCR	HCR
				WOS			
BFS	91/81	91/80	91/81	91/80	99.99	99.94	99.98
BFS-C	92/81	92/80	-	-	-	-	-
BT-BFS	91/80	91/80	91/80	91/80	99.95	99.89	99.94
DFS	91/81	91/80	91/81	91/80	100	99.89	99.98
DFS-C	91/82	91/81	-	-	-	-	-
BT-DFS	91/81	91/80	91/81	91/80	100	99.94	99.99
				BGC			
BFS	91/71/64/56	81/64/58/51	91/71/64/57	81/64/58/51	100/100/99.55	100/99.91/99.37	100/99.97/99.46
BFS-C	92/72/65/64	83/65/60/59	-	-	-	-	-
BT-BFS	91/70/64/63	80/63/58/55	91/70/64/63	80/63/58/55	99.99/99.97/100	99.99/99.95/100	99.99/99.96/100
DFS	92/71/65/60	82/64/58/52	92/71/65/60	82/64/58/52	100/100/100	100/100/100	100/100/100
DFS-C	92/73/67/65	82/66/61/57	-	-	-	-	-
BT-DFS	91/70/65/62	80/62/57/54	91/70/65/62	80/62/57/54	100/100/100	99.99/100/100	100/100/100
				RCV1V2			
BFS	94/80/84/13	92/65/62/13	94/80/84/13	92/65/62/13	100/99.99/100	100/99.75/100	100/99.95/100
BFS-C	94/80/83/30	92/65/56/30	-	-	-	-	-
BT-BFS	93/79/84/12	92/63/64/12	93/79/84/12	92/63/64/12	100/100/100	99.92/98.34/100	99.98/99.77/100
DFS	94/80/84/3	93/66/63/3	94/80/84/3	93/66/63/3	100/100/100	100/99.94/100	100/99.99/100
DFS-C	94/80/84/22	93/68/65/22	-	-	-	-	-
BT-DFS	93/79/84/22	92/65/63/22	93/79/84/22	92/65/63/22	100/100/100	99.83/99.95/100	99.97/99.99/100
DFS-Acr	94/79/83/03	92/62/57/03	94/79/83/03	92/62/57/03	100/99.99/100	100/99.93/100	100/99.98/100
DFS-C-Acr	94/79/83/03	92/62/60/03	-	-	-	-	-

Table 3. Per level results, for visibility, F1 scores are rounded to the nearest whole percentage, / is level separator, P-HCR, FP-HCR, HCR start from Level 2 since Level 1 can not violate the hierarchy.

4 Conclusion

In this work, we investigate three main design choices for HTC cast as a sequence-to-sequence task. First, we propose a complete framework, which can take any existing or future SHTC model and compare the impact of these different design choices on the architecture. This framework allows to validate that models performances stems from their intrinsic characteristics and not from different design choices from baselines. Second, through an empirical study of existing models, we devised interesting results for the practitioners: generating sequences from bottom to top yields more balanced predictions for each levels, generating sequence using DFS and use constrained generation is the best option overall, using label names instead of abstract tokens for classes have a significant impact on the performances of the system and invite to further investigate how to automatically select better sequence token for SHTC.

References

1. Bhambhoria, R., Chen, L., Zhu, X.: A simple and effective framework for strict zero-shot hierarchical classification. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 1782–1792. Association for Computational Linguistics (2023)
2. Cao, P., Chen, Y., Liu, K., Zhao, J., Liu, S., Chong, W.: HyperCore: Hyperbolic and co-graph representation for automatic ICD coding. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 3105–3114. Association for Computational Linguistics (2020)
3. Chen, H., Ma, Q., Lin, Z., Yan, J.: Hierarchy-aware label semantics matching network for hierarchical text classification. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing, ACL/IJCNLP. pp. 4370–4379. Association for Computational Linguistics (2021)
4. Chen, L., Chou, H.W., Zhu, X.: Developing prefix-tuning models for hierarchical text classification. In: Proceedings of Conference on Empirical Methods in Natural Language Processing: EMNLP. pp. 390–397. Association for Computational Linguistics (2022)
5. Deng, Z., Peng, H., He, D., Li, J., Yu, P.: HTCInfoMax: A global model for hierarchical text classification via information maximization. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 3259–3265. Association for Computational Linguistics (2021)
6. Giunchiglia, E., Lukasiewicz, T.: Coherent hierarchical multi-label classification networks. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) NeurIPS 2020, December 6-12, 2020, virtual (2020)
7. Huang, W., Liu, C., Xiao, B., Zhao, Y., Pan, Z., Zhang, Z., Yang, X., Liu, G.: Exploring label hierarchy in a generative way for hierarchical text classification. In: Proceedings of the 29th International Conference on Computational Linguistics. pp. 1116–1127. International Committee on Computational Linguistics (2022)
8. Jiang, H., Miao, Z., Lin, Y., Wang, C., Ni, M., Gao, J., Lu, J., Shi, G.: Financial news annotation by weakly-supervised hierarchical multi-label learning. In: Proceedings of the Second Workshop on Financial Technology and Natural Language Processing. pp. 1–7 (2020)
9. Jiang, T., Wang, D., Sun, L., Chen, Z., Zhuang, F., Yang, Q.: Exploiting global and local hierarchies for hierarchical text classification. In: Proceedings of EMNLP. pp. 4030–4039. Association for Computational Linguistics (2022)
10. Kowsari, K., Brown, D.E., Heidarysafa, M., Meimandi, K.J., Gerber, M.S., Barnes, L.E.: Hdltex: Hierarchical deep learning for text classification. In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA). pp. 364–371 (Dec 2017). https://doi.org/10.1109/ICMLA.2017.0-134

11. Lewis, D.D., Yang, Y., Russell-Rose, T., Li, F.: Rcv1: A new benchmark collection for text categorization research. Journal of machine learning research **5**(Apr), 361–397 (2004)
12. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net (2019), https://openreview.net/forum?id=Bkg6RiCqY7
13. Mao, Y., Tian, J., Han, J., Ren, X.: Hierarchical text classification with reinforced label assignment. In: Proceedings of EMNLP-IJCNLP. pp. 445–455. Association for Computational Linguistics (2019)
14. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research **21**(140), 1–67 (2020), http://jmlr.org/papers/v21/20-074.html
15. Rivas Rojas, K., Bustamante, G., Oncevay, A., Sobrevilla Cabezudo, M.A.: Efficient strategies for hierarchical text classification: External knowledge and auxiliary tasks. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 2252–2257. Association for Computational Linguistics (2020)
16. Shen, J., Qiu, W., Meng, Y., Shang, J., Ren, X., Han, J.: Taxoclass: Hierarchical multi-label text classification using only class names. In: Proceedings of NAACL-HLT. pp. 4239–4249. Association for Computational Linguistics (2021)
17. Silla, C.N.J., Freitas, A.x.: A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery **22**(1-2), 31–72 (2011)
18. Wang, Z., Wang, P., Huang, L., Sun, X., Wang, H.: Incorporating hierarchy into text encoder: a contrastive learning approach for hierarchical text classification. In: Proceedings of ACL. pp. 7109–7119. Association for Computational Linguistics (2022)
19. Yang, P., Sun, X., Li, W., Ma, S., Wu, W., Wang, H.: SGM: sequence generation model for multi-label classification. In: Bender, E.M., Derczynski, L., Isabelle, P. (eds.) Proceedings of COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018. pp. 3915–3926. Association for Computational Linguistics (2018)
20. Yu, C., Shen, Y., Mao, Y.: Constrained sequence-to-tree generation for hierarchical text classification. In: Proceedings of SIGIR. pp. 1865–1869. ACM (2022)
21. Zhou, J., Ma, C., Long, D., Xu, G., Ding, N., Zhang, H., Xie, P., Liu, G.: Hierarchy-aware global model for hierarchical text classification. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 1106–1117. Association for Computational Linguistics (2020)