Literature Review on Hierarchical Text Classification

Statistics Austria, Statistics Denmark

Hierarchical Text Classification (HTC) is a specialized form of text classification where classes are organized in a hierarchical structure. Silla and Freitas (2010) define hierarchical class structures as "IS-A relationships": They are asymmetrical, anti-reflective, and transitive; often represented as a tree where each node corresponds to a class label, with multiple possible children. As such, hierarchical classification concerns primarily formal, logical hierarchies in data structures.

Thus, the logical similarity (difference) between classes increases at lower (higher) hierarchical levels. If there were no hierarchy, all classes would be logically independent and unrelated, akin to a "flat" structure. In this view, HTC is widely applicable in domains of immediate interest to National Statistical Institutes (NSIs). It is especially relevant when classifying European standardized codes, such as ISCO, COICOP, PPP, NACE, ISCED etc. which represent hierarchical taxonomies. HTC aims to leverage the hierarchy to enhance classification accuracy.

Types of Hierarchical Text Classification Methods

Silla and Freitas (2010) define three types of approaches for classifying hierarchically structured labels: local, global and flattened. Flat classification problems ignore the class hierarchies and refer to traditional binary or multi-class classification. Hierarchically structured labels can be represented by a Direct Acyclic Graph (DAG) or tree with nodes connected by edges. As opposed to a tree, each node in a DAG can have more than one parent. The standardized codes systems that NSIs are working with are usually able to be represented by a tree, which is less difficult than working with DAGs. While the global approach uses only one model that considers all nodes of the hierarchy, the local approach, also known as top-down approach, makes use of multiple local classifiers, each learning only from a subset of nodes. These subsets can be divided into one classifier per node, one classifier per parent node and one classifier per hierarchy level. Naturally, this means that misclassifications on the higher levels are going to propagate to the lower levels. In contrast to the local classifiers, which in total consider all nodes in the tree, a flattened classifier trains only on the leaf nodes (i.e. the lowest level of the hierarchy), ignoring higher-level similarities and differences.

One advantage of the global classifier compared to the local classifier is that the global classifier only consists of one model, which usually results in a model of smaller size. This, however, usually results in the model being more complex than with local classifiers, as all hierarchy levels are predicted in the same step.

Foundational Approaches in Hierarchical Text Classification

Early approaches to HTC focused on traditional machine learning techniques, where hierarchical structures were incorporated through hierarchical constraints. In the following section, we summarize some of the pioneering research into hierarchical classification.

D’Alessio et al. (1998) investigate how the structure of a class hierarchy influences the performance of text classification systems. The authors compare flat classifiers, which treat categories independently, with hierarchical classifiers that utilize the relationships between categories organized in a taxonomy. Using a top-down classification approach, the classification process begins at the root of the hierarchy, where a classifier determines whether a document belongs to that general category. If the document is accepted at that level, the process continues down to the child nodes. At each level, a local classifier is trained to make decisions specific to that node’s subcategories. This means the model narrows down possible classifications step by step, guided by the structure of the hierarchy. By making decisions locally and conditioned on the previous predictions, the classifier leverages the hierarchical relationships to improve accuracy, particularly for categories with limited training data. Their findings indicate that hierarchical classifiers can outperform flat classifiers, particularly when the hierarchy accurately reflects semantic relationships among categories. By leveraging parent-child relationships, hierarchical classifiers can make more informed predictions, especially in cases where training data is sparse for specific subcategories.

Dumais et al. (2000) applied Support Vector Machines (SVMs) to hierarchical structures. Their model trained classifiers for each node in the hierarchy, a local classifier per parent node approach, where each classifier handled a specific subset of child categories. This approach demonstrated the advantages of HTC over flat classification, particularly in reducing classification error at higher levels of the hierarchy by leveraging hierarchical dependencies.

Ruiz and Srinivasan (2002) define the hierarchical classification as a divide and conquer approach, where a complex problem is solved by breaking it down into smaller, more manageable subproblems, solving each of those independently, and then combining their solutions to form a solution to the original problem. With this, they define smaller classification problems based on the underlying hierarchical structure. Their final classifier is an array of neural networks, more specifically supervised feedworward networks, that is ordered according to the hierarchy. The authors use medical text data to test their approach and find that compared to flat neural network classification the hierarchical model produces significantly better results.

Rousu et al (2006) present an approach to hierarchical text classification, where instances can belong to multiple categories simultaneously, i.e. a multi-label problem. The authors developed a kernel-based model using the Maximum Margin Markov Network framework, representing classification hierarchies as Markov trees to capture dependencies between labels. To ensure scalability, they propose an efficient optimization algorithm based on conditional gradient ascent with dynamic programming, allowing the model to handle thousands of training examples and large category hierarchies. The authors find that this method is as computationally efficient as training separate SVMs for each node, while providing better predictive performance by leveraging the structure of the hierarchy. This work significantly advances hierarchical multilabel classification by integrating structured label dependencies into the learning process.

Vens et al. (2007) introduce a method for constructing decision trees that can handle hierarchical multi-label classification, where each instance may be associated with multiple classes structured in a hierarchy. This work also considers hierarchies represented by DAGs (each node can have one or more parents), as opposed to most other work focusing only on the tree-like hierarchies (each node can only have one parent). Unlike flat or single-label approaches, their method directly incorporates the hierarchical relationships between classes into the decision tree induction process. The proposed algorithm uses a novel impurity measure that takes the hierarchy into account when selecting splits, allowing the model to better capture dependencies among labels. Experiments on real-world datasets demonstrate that this hierarchical approach improves predictive accuracy compared to traditional multi-label classifiers, especially when the class structure is complex.

These early HTC methods provided foundational insights and were able to show that incorporating the hierarchical structure of labels into the classification process improves prediction accuracies. However, they often faced challenges with computational efficiency and maintaining consistency in hierarchical predictions.

Advances in Neural-Network Models and Attention Mechanisms for HTC

With the advent of deep learning, researchers sought to improve HTC by using neural models to capture both text and label hierarchy structures. **Hierarchical Attention Networks (HANs)**, introduced by **Yang et al. (2016)**, became one of the influential architectures by using word- and sentence-level attention to capture document structure. Although initially developed for flat document classification, HAN's hierarchical structure inspired many HTC models due to its ability to capture multi-level information within a document.

Recent years have seen significant developments in HTC methodologies, driven by advancements in machine learning and natural language processing. **Wehrmann et al. (2018)** proposed a model called **Hierarchical Multi-Label Classification Networks (HMCN)**, that uses a hybrid loss function that captures both global and local class relationships within the hierarchy and penalizes hierarchical violations. This approach is especially suitable for multi-label classification in hierarchical label settings.

The paper **Hierarchical Multi-label Classification of Text with Capsule Networks (HMC-Capsule)** by **Aly et al. (2019)** explores the use of capsule networks for hierarchical multi-label text classification, a task where a document can belong to multiple categories structured in a hierarchy. The authors compare capsule networks against conventional methods like CNNs, LSTMs, and non-neural approaches such as SVMs. The results demonstrate that capsule networks outperform traditional models, especially for rare labels and complex hierarchical structures. The strength of capsule networks lies in their ability to capture and combine latent information effectively, which is particularly advantageous in scenarios with sparse or imbalanced data. The term latent information, or latent variables, refers to information not directly observable but inferred from observable data. It represents underlying hidden factors that explain patterns in the observed data. In the context of deep learning, the latent information represents internal representations that the network learns to capture during training. These representations are not explicitly labeled in a dataset but are crucial for the model to make accurate predictions. Capsules networks use this latent information in the form of vectors instead of a single scalar value like traditional neural networks do.

HiAGM (Hierarchy-Aware Global Model) proposed by **Zhou et al. (2020)** is a hierarchy-aware model that introduces structure encoders to model label dependencies. HiAGM allows the model to capture complex label interactions and dependencies in deep hierarchies, an approach that significantly outperforms traditional hierarchical classification in terms of both accuracy and interpretability. Graph-based HTC models are particularly well-suited for tasks with highly structured taxonomies, as they can naturally handle label relationships without manually engineered constraints.

Similarly, the **HiMatch (Hierarchy-Aware label semantics Matching network)** by **Chen et al. (2019)** uses a text encoder in combination with a label encoder, that generates label embeddings that capture and enhance the dependencies between labels. For the initial label vector, the pre-trained BERT embeddings are used. They define the classification problem as a semantic matching problem, where the relationship between text and label embeddings is modeled by a joint embedding loss and a matching learning loss. The authors observe that HiMatch is superior to BERT and HiAGM, as well as other state-of-the-art hierarchy aware models. They also report that a model where the HiMatch text encoder is replaced with BERT achieves the best results.

Yu et al. (2022) introduce a sequence-to-tree (Seq2Tree) framework, which, instead of treating hierarchical classification as a flat multi-label classification task, generates labels as a sequence in tree structured manner that directly reflects the hierarchy. This approach outperformed other state-of-the-art methods, including HMC-Capsule, HiAGM and HiMatch, on three benchmark data sets, demonstrating higher classification performance.

The paper "Incorporating Hierarchy into Text Encoder: a Contrastive Learning Approach for Hierarchical Text Classification" by **Wang et al. (2022)** introduces **Hierarchy-Guided Contrastive Learning (HGCLR)**. The idea is to enhance performance by making text representations inherently aware of hierarchical structures. To do so the method embeds hierarchical label structures directly into text encoders such that the classes/labels are also modelled as vectors (embeddings), allowing the text and labels to be passed along combined. This deviates from typical classification methods that process text and label hierarchies separately. As a contrastive learning method, it does not immediately predict a class with some probability. Instead, it first increases distances between pairs (of different label and text embeddings) that are unrelated and decreases the distance between pairs that are similar. Therefore, contrasting is also referred to as “pushing” and “pulling” apart the pairs. The different pairs are dubbed “positive” and “negative” samples. Positive samples consist of the input text and the label embeddings of the true label, its ancestors and possible children. “Negative samples”, on the other hand, consist of the input text and the embeddings of unrelated labels, i.e. labels that are not in the path of the true label. Thus, positive samples are guided by label hierarchies, enabling the text encoder to generate hierarchy-aware representations (without relying on separate label encodings). On this basis, multi-label classification on a flattened hierarchy is performed. The authors show superior performance of the HGCLR on three data sets when compared to the performance of other state-of-the-art models like BERT and HiMatch.

Based on the HGCLR model **Zhang et al. (2024)** propose the **Hierarchy-Aware and Label Balanced Model (HALB)** for Hierarchical Text Classification, with the aim to create a model with even more hierarchical awareness. They propose two new aspects to the HGCLR: firstly, a method called multi-label negative supervision drives representations of samples with very different labels further apart. The author’s idea is that this will solve the problem of semantically similar texts having different labels, which is a problem faced by the HGCLR . Secondly, they adopt asymmetric loss instead of cross-entropy loss for classification loss to deal with label imbalance. With the asymmetric loss, the model is supposed to focus on samples that are more difficult to predict. They claim that this idea works because the positive and negative labels tend to contribute similarly to the loss function. Compared to the strongest baseline model, the HGCLR, the HALB Model achieves an average improvement of 0.51% and 1.28% for Micro- and Macro-F1 respectively.

Evaluation Metrics

Whether HTC is worthwhile is of particular interest because it is more taxing (technically, computationally, etc.). In the broader field of classification and beyond, it is standard to use criteria or metrics for evaluating performance. If HTC performs better, the method is worthwhile. Interestingly, however, Sun and Lim (2001) suggested that criteria typically used to assess classification problems—accuracy, F1, precision, and recall—all favor flat classification models. That is, these metrics are biased because they discount whether some errors are more severe for classification than others; something that arguably follows directly from the assumption of hierarchical class structures. In the presence of hierarchy, misclassifications at higher levels are supposedly more inaccurate as compared to misclassifications at lower levels. Classifying fruits as animals (e.g. “orange” as “cat”) is more inaccurate than classifying fruits (e.g. “orange” as “apple”). In other words, a prediction that is closer to the true class is less severe than a prediction that is distant from the class. HTC may thus increase performance more than is typically believed from evaluating the usual criteria.

To solve this problem, **Sun and Lim (2001)** proposed evaluating classification against class-similarity and class-distance. While there are many ways to calculate similarity, the authors decide on the cosine distance. Curiously, the authors propose calculating similarity of class pairs based on their feature vectors (i.e. those used to predict classes or classify observations), not the class hierarchy in and of itself. Yet, these are summed over all observations for a given pair. On this basis, the average class similarity can be computed, and the authors propose using it to evaluate the degree of classification correctness. Only when classes are incorrectly predicted do this class-similarity criterion diverge from the usual “flat” performance criteria. In cases of misclassification, the proposed metric can express “how much” a predicted class contributes to the true class; directly tapping into the idea that some misclassifications are more inaccurate than others. Briefly put, the authors propose to measure class-distance as the number of links between the class pairs under evaluation.

More recent work by **Kosmopoulos et al. (2015)** introduces precision, recall and F1 measures based on the concept of the lowest common ancestor (LCA), proposed by **Aho et al. (1973)**. Given a tree, the Lowest Common Ancestor of two nodes u and v is the deepest node (i.e., farthest from the root) that is an ancestor of both u and v. With the set of true labels Y , and the set of predicted labels $\overline{Y}$, Kosmopoulos et al. (2015) define

$$precision_{LCA} = \frac{|\overline{Y}_{aug} \cap Y_{aug}|}{|\overline{Y}_{aug}|},$$

$$recall_{LCA} = \frac{|\overline{Y}_{aug} \cap Y_{aug}|}{|Y_{aug}|}$$

$$F1_{LCA} = \frac{2 * precision_{LCA} * recall_{LCA}}{precision_{LCA} + recall_{LCA}}.$$

Here, $\overline{Y}_{aug}$ and Y_{aug} denote an augmented set, which includes the labels themselves (Y or $\overline{Y}$), their ancestors, and the LCAs between each predicted and true label. These three measures are especially relevant for hierarchical labels, as they take partial correctness into account and do not treat the classes as flat label set.

In any event, to better understand the contribution of hierarchical evaluation metrics, Plaud et al. (2024) introduced a new, challenging dataset, the Hierarchical WikiVitals (HWV), accompanying three common ones (WOS, RCV1, BGC). Though not stated explicitly, “challenging” in this sense seems to imply a more hierarchical structure of six levels of depth. If anything, HTC is expected to score better performance than “flat” classification on such data using hierarchical evaluation metrics. Across the four datasets, Plaud and colleagues reported four evaluation metrics, including micro and macro F1 as well as hierarchical F1: hF1. Micro F1 weighs each observation equally regardless of its class. Macro F1 weighs each class equally regardless of its number of observations, implying that macro F1 penalized class imbalances. And the authors notice that class imbalance is strongly correlated with hierarchical depth (the number of levels). Indeed, half the class labels in HWV have <10 observations. With regard to their results, the performance difference was smallest on WOS, which has only two hierarchical levels, and largest on HWV (RCV1 and BGC both have four levels). Moreover, micro F1 and hF1 scores were less different than macro F1 and hF1 on all datasets. In all cases, hF1 evaluated the performance of the methods better than the other evaluation metrics. In conclusion, Plaud and colleagues suggest (page 239): “when used on a more challenging dataset, state-of-the-art hierarchy-aware HTC models are less able to integrate that complex hierarchical information into their prediction than a simple model trained with conditional softmax cross-entropy". As such, the most striking take-away arguably is that their proposed loss function seems to perform better on hierarchical data structures because it better handles the implied class imbalance (see e.g. Figure 4). To be clear, this requires assuming that micro F1 and hF1 are indeed the best evaluation criteria rather than inflated or biased metrics, as a critical assessment might suggest.

Challenges and Future Directions

While HTC has made remarkable strides, challenges remain, especially in balancing accuracy and efficiency in large and deep hierarchies. Hierarchical inconsistencies, where a model predicts child classes without their corresponding parent classes, remain a prominent issue, as traditional HTC methods often lack built-in mechanisms to enforce hierarchical constraints.

Another ongoing challenge is datasets with class imbalance, meaning that some classes are underrepresented. Hierarchical datasets frequently exhibit extremely skewed long-tailed distributions where certain classes are observed much less. This leads to models that are worse at predicting less observed classes. To be sure, this problem is not unique to HTC but remains in “flat” classification tasks, albeit that does not make the problem disappear. Potentially, it can be resolved by using hierarchy-aware data augmentation, class weights or resampling methods. Resampling strategies have already been proposed, however, in related fields deviating from the strict sense of HTC, that is, few-shot meta-learning Siamese networks (Han et al., 2023). Coupled with the conclusion by Plaud et al. (2024), this could be taken to indicate that class imbalance is at the core of the challenge confronting HTC (rather than the hierarchy per se) and something better solved by alternative frameworks targeted at solving that specifically. There are other possibilities, too. Chen et al. (2024) proposed ‘in-context’ learning, that is to say, prompting to LLMs information about the task along information about the hierarchical data structure in order to generate more label description. Although interesting, their proposed solution was tested on hierarchies 2-4 deep only (one of which, to be sure, has 300 unique classes, though their research focused on the simplest, WOS). A somewhat similar approach called TELEClass was offered by Zhang et al. (2025), suggesting passing “class-specific” or “class-indicative” terms generated from the corpus to the model.

In the introduction, we mentioned the formal definition of hierarchy which serves as the point of departure for HTC. Furthermore, previous work has introduced performance metrics that account for (average) class similarity when assessing the correctness of classification (in the context of contrastive learning, e.g. Wang et al., 2022; Yu et al., 2024). In our view, the concept of class similarity poses a challenge to classifying formal hierarchies or, in other words, suggests why HTC is challenging. Whether formal definitions map onto empirical structures may well be at the core of the problem. In the literature about hierarchical, or multi-level, regression models it is normal practice to explicitly estimate the extent of hierarchy or clustering in the data (e.g. Gelman & Hill, 2007). In other words, the degree of hierarchy is assessed a priori (some unit of analysis is nested within another larger unit of analysis) and then empirically assessed by the statistical workhorse known as the Intra-Class Correlation. The concept has already been utilized for feature selection in hierarchical classification (Shi et al., 2023), discussed in regard to contrastive learning on generated samples (Yu et al., 2023), and few-shot learning (Han et al., 2023). Like the running assumption throughout the present review, HTC is likely to excel over "flat" classification the more strongly formal class hierarchies are undergirded empirically. That is, texts assigned to class L1_a > L2_a (e.g., *oats*) should be more similar to each other than to those of class L1_a > L2_b (e.g., *wheat*), while those two classes also should be more similar than texts belonging to sibling classes such as L1_b (e.g., *legumes*) or L1_c (e.g., *nuts*). In other words, the more the empirical structure of the classes corresponds to the formal hierarchical taxonomy, the more leverage HTC will provide over the classification problem compared to a "flat" classification model. Moreover, predictors or features observed at different "logical" hierarchical levels, thus being constant at lower or deeper levels, may be of particular interest. So, rather than using class similarity to evaluate the performance of HTC, perhaps it can be embedded into HTC or developed to discern whether to use HTC in the first place.

Given this situation, researchers (perhaps more than applied users in Official Statistics Bureaus) may consider designing a study comparing multiple datasets varying in the properties of hierarchies (e.g. depth, class imbalance, etc.), using various evaluation criteria varying in their account of hierarchy, and training models some of which are HTC and others which are "flat".

With growing model and data complexity comes the challenge of scalability and computational costs. Large and complex amounts of data require complex models, which are expensive to train. Therefore, there is a need for efficient algorithms that enable training such data.

Yet, perhaps more pressingly, actual implementations—meaning software to use on a computer given some hierarchical data—are needed in the first place. Zangari et al. (2024) recently commented that many proposed methods are not accompanied by public domain, readily available implementations. Thus, even though the current review may give an impression that research develops at a fast pace, this does not necessarily mean the methods are easily accessible for Official Statistics. In many cases, in-house development may be needed, and thus collaborations like AIML4OS may seem a needed opportunity to co-develop and share methodology and their implementation.

Conclusion

HTC research has evolved significantly, from early SVM-based methods to sophisticated neural architectures that leverage hierarchical structures in both labels and text. Recent advances in transformer models, label embeddings, and graph-based approaches have pushed the boundaries of HTC, enabling models to handle deeper hierarchies and complex dependencies. However, challenges such as hierarchical consistency, computational efficiency, and adaptability to evolving taxonomies remain open research areas. Continued research in HTC, particularly with hybrid models that integrate external knowledge and hierarchy-aware mechanisms, will be essential for advancing the field and expanding HTC applications across various domains.

At Statistics Austria we are currently experimenting with local hierarchical text models, such as one transformer model per hierarchy. Ideally, we would also like to implement a global classifier, similar to the ones mentioned above, if resources allow it. However, we have not decided on how this is to be implemented. We are planning on comparing any HTC models with our baseline flat classification transformer model using the above-mentioned hierarchy aware evaluation metrics. Currently, we are only focusing on language models and will most likely not compare the results for traditional machine learning models of flat, local and global classifiers.

At Statistics Denmark we are focusing on getting a working-model into production. The results for our Official Statistics problem thus far do not strongly favor prioritizing further development of advanced classification models in frameworks like PyTorch and BERT fine-tuning across hierarchical levels (e.g. Bouchiha et al., 2025; Devlin et al., 2019) vis-a-vis implementing the simpler framework of FastText. In that regard, our results compare with Stein et al. (2018). We are dealing with relatively scarce, deep, and imbalanced hierarchical data whose texts are short. That is, about 10.000 records (in that order of magnitude) scattered over roughly 400 classes at the deepest level, 7 levels deep, with string lengths less than 30. Fine-tuning BERT requires many observations per class, and the highest level in the hierarchy (level 2) has the most observations per class but also provides the least information about the hierarchy. Fine-tuning--whether used for global classification, LCPN or LCN—is thus confronted with the problem that class imbalance gets more severe as hierarchies deepen. We are currently heading toward rethinking the analysis of the business problem or its first principles, possibly leading us to abandon hierarchical classification as it is strictly defined. When a given item has already been labelled or classified, whether by manual supervision or automated, there is no need to do that again because the item is uniquely identified, leaving a set-theoretic problem for SQL-like computations, not ML/AI. Our main Official Statistics problem is chiefly about new or unseen items thus far being handled manually (used as input to train classification). It concerns us whether even an excellent classification model will ever stand a chance to help alleviate manual labor used, given the data input we can expect. Such prospects head our thinking in the direction of considering exploring few- or zero-shot learning directly on the authoritative taxonomy or schema developed in the context of Eurostat (besides on the unlabeled data). This could mean prompting GPTs (transformers, transfer learning) and vectorizing each class document (e.g. Brown et al., 2020; Chen et al., 2024; Zhang et al., 2025) or developing Siamese networks, or meta-learning, using e.g. contrastive loss (Han et al., 2023; Neculoiu et al., 2016; Wang et al., 2022; Yang et al., 2020; Yu et al., 2023). However, this comment should be taken as ideas or opinions in-progress more than action plans, as developing such models will require us to devote additional resources.

References

Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1973, April). On finding lowest common ancestors in trees. *In Proceedings of the fifth annual ACM symposium on Theory of computing (pp. 253-265).*

Aly, R., Remus, S., & Biemann, C. (2019, July). Hierarchical multi-label classification of text with capsule networks. In *Proceedings of the 57th annual meeting of the association for computational linguistics: student research workshop* (pp. 323-330).

Bouchiha, D., Bouziane, A., Doumi, N., Hamzaoui, B., & Boukli-Hacene, S. (2025). Hierarchical Text Classification: Fine-tuned GPT-2 vs BERT-BiLSTM. *Applied Computer Systems*, *30*(1), 40-46.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877-1901.

Chen, H., Ma, Q., Lin, Z., & Yan, J. (2021, August). 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)* (pp. 4370-4379).

Chen, H., Zhao, Y., Chen, Z., Wang, M., Li, L., Zhang, M., & Zhang, M. (2024). Retrieval-style in-context learning for few-shot hierarchical text classification. *Transactions of the Association for Computational Linguistics*, *12*, 1214-1231.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). 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)* (pp. 4171-4186).

D'Alessio, S., Murray, K. A., Schiaffino, R., & Kershenbaum, A. (2000, April). The Effect of Using Hierarchical Classifiers in Text Categorization. In *Riao* (pp. 302-313).

Dumais, S. a. (2000). Hierarchical classification of web content. *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval.*

Gelman, A., & Hill, J. (2007). *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press.

Han, C., Wang, Y., Fu, Y., Li, X., Qiu, M., Gao, M., & Zhou, A. (2023, April). Meta-learning siamese network for few-shot text classification. In *International Conference on Database Systems for Advanced Applications* (pp. 737-752). Cham: Springer Nature Switzerland.

Kosmopoulos, A., Partalas, I., Gaussier, E., Paliouras, G., & Androutsopoulos, I. (2015). Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Mining and Knowledge Discovery, 29, 820-865.*

Neculoiu, P., Versteegh, M., & Rotaru, M. (2016, August). Learning text similarity with siamese recurrent networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP* (pp. 148-157).

Plaud, R., Labeau, M., Saillenfest, A., & Bonald, T. (2024). Revisiting Hierarchical Text Classification: Inference and Metrics. *Proceedings of the 28th Conference on Computational Natural Language Learning*, Miami, United States. pp.231-242.

Rousu, J., Saunders, C., Szedmak, S., & Shawe-Taylor, J. (2006). Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research*, *7*, 1601-1626

Ruiz, M. E., & Srinivasan, P. (2002). Hierarchical text categorization using neural networks. *Information retrieval*, *5*, 87-118.

Shi, J., Li, Z., & Zhao, H. (2023). Feature selection via maximizing inter-class independence and minimizing intra-class redundancy for hierarchical classification, *Information Sciences*, 626, 1-18.

Silla, C. N., & Freitas, A. A. (2011). A survey of hierarchical classification across different application domains. *Data mining and knowledge discovery*, *22*, 31-72.

Stein, R. A., Jaques, P. A., & Valiati, J. F. (2019). An analysis of hierarchical text classification using word embeddings. *Information Sciences*, *471*, 216-232.

Sun, A., & Lim, E. P. (2001, November). Hierarchical text classification and evaluation. In *Proceedings 2001 IEEE International Conference on Data Mining* (pp. 521-528). IEEE.

Vens, C., Struyf, J., Schietgat, L., Džeroski, S., & Blockeel, H. (2008). Decision trees for hierarchical multi-label classification. *Machine learning*, *73*, 185-214.

Wang, Z., Wang, P., Huang, L., Sun, X., & Wang, H. (2022). Incorporating hierarchy into text encoder: a contrastive learning approach for hierarchical text classification. *arXiv preprint arXiv:2203.03825*.

Wehrmann, J., Cerri, R., & Barros, R. (2018, July). Hierarchical multi-label classification networks. In *International conference on machine learning* (pp. 5075-5084). PMLR.

Zangari, A., Marcuzzo, M., Rizzo, M., Giudice, L., Albarelli, A., & Gasparetto, A. (2024). Hierarchical Text Classification and Its Foundations: A Review of Current Research. *Electronics*, *13*(7), 1199. doi.org/10.3390/electronics13071199

Zhang, J., Li, Y., Shen, F., Xia, C., Tan, H., & He, Y. (2024). Hierarchy-aware and label balanced model for hierarchical text classification. *Knowledge-Based Systems*, *300*, 112153.

Zhang, Y., Yang, R., Xu, X., Li, R., Xiao, J., Shen, J., & Han, J. (2025, April). Teleclass: Taxonomy enrichment and llm-enhanced hierarchical text classification with minimal supervision. In *Proceedings of the ACM on Web Conference 2025* (pp. 2032-2042).

Yang, W., Li, J., Fukumoto, F., & Ye, Y. (2020, November). HSCNN: a hybrid-siamese convolutional neural network for extremely imbalanced multi-label text classification. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)* (pp. 6716-6722).

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016, June). 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* (pp. 1480-1489).

Yu, C., Shen, Y., & Mao, Y. (2022, July). Constrained sequence-to-tree generation for hierarchical text classification. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval* (pp. 1865-1869).

Yu, S., He, J., Gutiérrez-Basulto, V., & Pan, J. Z. (2023). Instances and labels: Hierarchy-aware joint supervised contrastive learning for hierarchical multi-label text classification. *arXiv preprint* :2310.05128.