Revisiting Hierarchical Text Classification: Inference and Metrics

Anonymous ACL submission

Abstract

Hierarchical text classification (HTC) is the task of assigning labels to a text within a structured space organized as a hierarchy. Recent works treat HTC as a conventional multilabel classification problem, therefore evaluating it as such. We instead propose to evaluate models based on specifically designed hierarchical metrics and we demonstrate the intricacy of metric choice and prediction inference method. We introduce a new and challenging HTC dataset and we evaluate fairly recent sophisticated models, comparing them with a range of simple but strong baselines. Finally, we show that those baselines are very often competitive with the latest HTC models. Our works shows the importance of carefully considering the evaluation methodology when proposing new methods for HTC.

1 Introduction

Text classification is a long-studied problem that may involve various types of label sets. In particular, Hierarchical Text Classification (HTC) involves labels that exhibit a hierarchical structure with parent-child relationships. The structure that emerges from these relationships is either a tree (Kowsari et al., 2018; Lewis et al., 2004; Lyubinets et al., 2018; Aly et al., 2019; Sandhaus, 2008) or a Directed Acyclic Graph (DAG) (Bertinetto et al., 2020). Each input example then comes with a set of labels that form one or more paths in the hierarchy. A first crucial challenge in HTC lies in accurately evaluating model performance. This requires metrics which are sensitive to the severity of prediction errors, penalizing mistakes with larger distances within the hierarchy tree. While pioneering efforts have been made by Kiritchenko et al. (2006), Silla and Freitas (2011), and Kosmopoulos et al. (2014), evaluation in the context of hierarchical classification remains an ongoing research area.

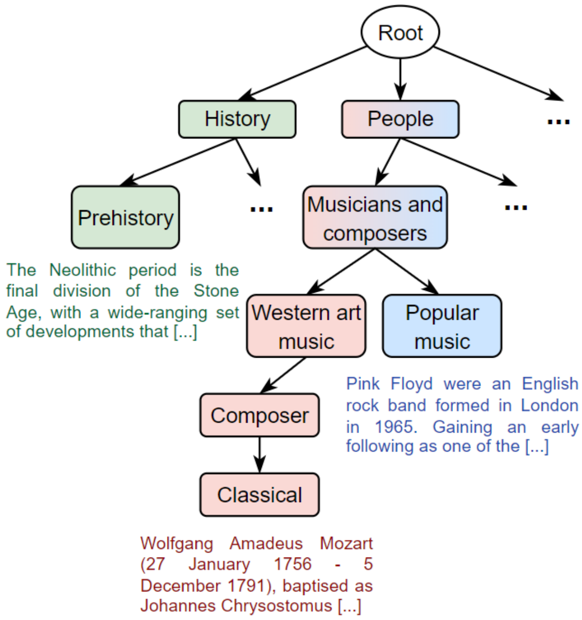

Figure 1: Extract of the taxonomy of our new dataset Hierarchical WikiVitals. Each colored path in the tree is the set of labels of the input text of the same color.

There is a substantial body of literature addressing HTC. The most recent methods produce text representations which are *hierarchy-aware*, as they integrate information about the label hierarchy (Song et al., 2023; Zhou et al., 2020; Deng et al., 2021; Wang et al., 2022b,a; Jiang et al., 2022; Chen et al., 2021; Zhu et al., 2023). However, we believe that evaluation with these models have been insufficiently investigated. In this work, we plan to shed light on inference strategies – the way of producing predictions, given a probability distribution over the nodes of the hierarchy – which we consider an under-addressed challenge. We provide new insights, emphasising the intricacy of inference and evaluation, which cannot be considered separately. To complete this investigation, we introduce a new English benchmark dataset, Hierarchical Wikivitals (HWV), which we intend to be significantly more challenging than the usual HTC benchmarks in English (see Figure 1 for an extract of the taxonomy). Finally, we experiment within our proposed framework, verifying the performance of recent models against simpler ones, which rely solely on a text encoder (Devlin et al., 2019) and a loss function (Bertinetto et al., 2020; Vaswani et al., 2022; Zhang et al., 2021) able to integrate local hierarchical information, such as the conditional softmax and sigmoid. To summarize, our contributions are:

1. We propose to quantitatively evaluate HTC methods based on specifically designed hierarchical metrics.
2. We prove that inference is often not tailored to the metrics used, and we therefore propose an adapted evaluation methodology.
3. We present a novel HTC dataset, Hierarchical WikiVitals, equipped with a complex and challenging label hierarchy.
4. We provide a rationale for the adoption of the conditional softmax and conditional sigmoid as strong baselines for the task, establishing a theoretical connection between them.
5. Our experiments reveal that simple models are very often competitive with sophisticated ones when properly evaluated.

Problem definition

Hierarchical Text classification (HTC) is a subtask of text classification which consists in assigning to an input text $x \in \mathcal{X}$ a set of labels $Y \subset \mathcal{Y}$, where the label space $\mathcal{Y}$ exhibits parent-child relationships. We call hierarchy the directed graph $\mathcal{H} = (\mathcal{Y}, \mathcal{E})$, where $\mathcal{E} \subset \mathcal{Y}^2$ is the set of edges, which goes from a parent to its children. We restrain our study to the case where $\mathcal{H}$ is a tree. We follow the notations of Valmadre (2022) and call $\mathbf{r} \in \mathcal{Y}$ the unique root node and $\mathcal{L}$ the set of leaf nodes. For a node $y \in \mathcal{Y} \backslash \{\mathbf{r}\}$ we denote $\pi(y)$ its unique parent, $\mathcal{C}(y) \subset \mathcal{Y}$ the set of its children and $\mathcal{A}(y)$ the set of its ancestors (defined inclusively). A label set Y of an input x cannot be arbitrary: if $y \in Y$ then, due to the parent relations, we necessarily observe that $\mathcal{A}(y) \subset Y$. An even more restrictive framework is the *single-path leaf labels* setting, where (1) Y is a single path in the tree: $y_1, y_2 \in Y \Rightarrow y_1 \in \mathcal{A}(y_2)$ or $y_2 \in \mathcal{A}(y_1)$, and (2) Y reaches a leaf: $Y \cap \mathcal{L} \neq \emptyset$. As Valmadre (2022), we study methods that map an input text x to a conditional distribution $\mathbb{P}(\cdot|x)$ over $\mathcal{Y}$, whose estimation is denoted $\hat{\mathbb{P}}(\cdot|x)$.

2 Related Work

2.1 Hierarchical Text Classification

Hierarchical classification problems, including the particular case of HTC, are typically dealt with through either a *local* approach or a *global* one. We refer to the original definition made by Silla and Freitas (2011), according to which the difference between the two categories lies in the training phase. Indeed, local methods imply training a collection of specialized classifiers, *e.g.* one for each node, for each parent node or even one for each level; and during its training each classifier is unaware of the holistic structure of the hierarchy (Zangari et al., 2023). While often computationally costly, it has proven to be effective to capture crucial local information. Along those lines, Banerjee et al. (2019) propose to link the parameters of a parent classifier and those of its children, following the idea of transferring knowledge from parent nodes to their descendants (Shimura et al., 2018; Huang et al., 2019; Wehrmann et al., 2018). Besides their cost, local approaches have the issue of potential exposure bias, as decisions are taken without access to information about the whole structure.

Conversely, global methods involve a unique model that directly incorporates the whole hierarchical information in their predictions. There exist very different types of global approaches, from which we can draw two broad categories: losses incorporating hierarchical penalties and hierarchy-aware models.

Hierarchical penalties. The idea of these methods is generally to use a standard binary cross-entropy (BCE), and add penalisation terms that incorporate hierarchical information. Gopal and Yang (2013) and Zhang et al. (2021) propose regularization based on hypernymy, either acting on on the parameter space or the outputted probability space, while Vaswani et al. (2022) introduce an enhanced BCE loss, named CHAMP, which penalises false positives based on their distance to the ground truth in the hierarchy tree.

Hierarchy-aware models. In order to include the structural constraints of the hierarchy to the prediction, Mao et al. (2019) propose a reinforcement learning approach, and Aly et al. (2019) an architecture based on capsule networks. But recent works obtained state-of-the-art results by combining a text encoder with a structure encoder applied to the label hierarchy: this idea was first proposed by Zhou et al. (2020), using graph convolution networks as hierarchy encoder. Based on this seminal work, Jiang et al. (2022) separately incorporate local and global hierarchy information, and Wang et al. (2022a) propose a contrastive learning approach, while Zhu et al. (2023) implement a method to encode hierarchy with the guidance of structural entropy, following many previous works on the idea (Chen et al., 2020; Zhang et al., 2022; Deng et al., 2021; Chen et al., 2021; Wang et al., 2021). We should note that these models are usually trained with a BCE loss (or one of its penalized version (Zhang et al., 2021)).

2.2 Hierarchical prediction

Making a prediction in HTC involves two seemingly irreconcilable difficulties: *prediction coherence* and *error propagation*. Typically, one has to decide between making independent predictions, which may lead to coherence issues (e.g., predicting a child without predicting its parent), or employing a top-down inference approach, which may cause error propagation issues (Yang and Cardie, 2013; Song et al., 2012). This trade-off is arbitrated by the choice of the modelisation: a global BCE-based loss may produce incoherent predictions while local structure-aware losses (Redmon and Farhadi, 2017; Bertinetto et al., 2020) can lead to exposure bias. Recent hierarchy-aware models predominantly operate within the former framework, training and evaluating the model as a simple multi-label classifier, at the price of ignoring potentially badly structured predictions.
In this work, we propose to revisit this trade-off by improving our *evaluation framework*. We will experiment with recent hierarchy-aware models, hierarchical penalties, but also, top-down loss-based approaches.

2.3 Hierarchical classification evaluation

In the context of HTC, inference is mostly performed through thresholding to 0.5 the estimated probability distribution over nodes, and computing the F1-score (*micro* and *macro*), which amounts to multi-label evaluation. However, a lot of efforts have already been dedicated to proposing metrics within a hierarchical context: *hierarchical metrics*. The underlying idea is simple: take into account the severity of an error based on the known hierarchy: predicting a *Bulldog* instead of a *Terrier* should be less penalized than predicting a *Unicorn* instead of a *Terrier*. This has been extensively studied in Kosmopoulos et al. (2014). The first intuitive way to deal with this, is to compute a shortest-path (SP). Roughly, it corresponds to computing the number of edges between a predicted node and the ground truth one. Depending on assumptions we make, it may be ill-defined, especially when there are multi-path labels (Kosmopoulos et al., 2014). But in a simple *single-path leaf label* setting, it yields an interpretable metric. Efforts were also made to adapt metrics used in a standard multi-label classification problem to a hierarchical context. This motivated the Hierachical Recall, Precision and F1-scores (Kiritchenko et al., 2006; Kosmopoulos et al., 2014) which imply predicting the full path: *Bulldog, Dogs, Animals* and *Unicorn, Animals* rather than *Bulldog* and *Unicorn*. Looking at which part of the path is well predicted then allows to take into account the severity of errors. In a standard multi-label framework these metrics are often computed at different operating points, thus yielding a trade-off curve. To our knowledge only Valmadre (2022) proposed such an evaluation methodology in a hierarchical context. In this work, we choose to use the shortest path and hierarchical F1-score for evaluation. In order for SP to be properly defined, we choose as main setting for our experiments the *single-path leaf labels* framework, which we will then extend to multi-path labels.

3 Evaluation metrics

3.1 Hierarchical metrics

We begin by detailing the two hierarchical metrics we will work with in our experiments. Formally, suppose that, given $\hat{\mathbb{P}}(\cdot|x)$, we obtain $\hat{Y}$ the predicted set of labels, which we confront to the ground truth Y. A prediction is called *coherent* if $z \in \hat{Y} \Rightarrow \mathcal{A}(z) \subset \hat{Y}$.

Shortest Path. We define the shortest path metric (Garnot and Landrieu, 2021) $\mathrm{SP}(Y, \hat{Y})$ as the length of shortest path in $\mathcal{H}$[1] between the most specific element of Y denoted y^{spe} and the most specific element of $\hat{Y}$ denoted $\hat{y}^{\text{spe}}$[2], which we would like to minimize. Little consideration was given to this metric in the literature, although it provides very intuitive and interpretable results.

Hierarchical F1-score. Introduced by Kiritchenko et al. (2006), it consists in augmenting

[1] In which we undirected the edges

[2] Metric definition implicitly supposes $\hat{Y}$ is a single path

$\hat{Y}$ with all its ancestors as follows :

$$\hat{Y}^{\text{aug}} = \bigcup_{\hat{y} \in \hat{Y}} \mathcal{A}(\hat{y}) \tag{1}$$

And to compute the hierarchical precision, recall and F1-score as follows :

$$\text{hP}(\text{Y}, \hat{\text{Y}}) = \frac{\left|\hat{Y}^{\text{aug}} \cap Y\right|}{\left|\hat{Y}^{\text{aug}}\right|} \quad \text{hR}(\text{Y}, \hat{\text{Y}}) = \frac{\left|\hat{Y}^{\text{aug}} \cap Y\right|}{|Y|}$$

$$\text{hF1}(\text{Y}, \hat{\text{Y}}) = \frac{2 \cdot \text{hP}(\text{Y}, \hat{\text{Y}}) \cdot \text{hR}(\text{Y}, \hat{\text{Y}})}{\text{hP}(\text{Y}, \hat{\text{Y}}) + \text{hR}(\text{Y}, \hat{\text{Y}})}$$

In the multi-label setting, there are several methods of aggregation to compute a global F1-score.[3]. We define here a per-instance hF1-score as per Kosmopoulos et al. (2014) which is then averaged over all inputs (referred as *samples* setting). In its very first introduction, it was defined in a *micro* fashion by Kiritchenko et al. (2006) (see Appendix B.2 for full definitions).

Proposition 1 *In micro and samples settings, if every prediction $\hat{Y}$ is coherent then hF1 and F1 are strictly equal.*

Proof is detailed in Appendix B.2. It was therefore relevant to employ the *micro* F1-score as it is done in recent literature: when predictions are coherent, it is indeed a hierarchical metric.

3.2 Inference methodology

In this section, we argue against the practice of using a BCE-based loss and a threshold set to 0.5 to produce predictions. While this corresponds to minimizing the Hamming loss in case of label independence (Dembczyński et al., 2012), to the best of our knowledge, there is no evidence of the optimality of such a predictor in a hierarchical setting.

3.2.1 Risk Minimization

Risk minimization is a long-time studied topic (Vapnik, 1999), addressing the problem of finding an optimal predictor f^* while optimizing a metric L. Re-writing this minimization yields the *Bayes-Optimal predictor*:

$$f^*(x) = \underset{\hat{Y}}{\text{argmin}}\ \mathbb{E}[L(Y, \hat{Y})|X = x] \tag{2}$$

When Equation (2) has a closed-formed solution, this gives a predictor which optimizes metric L.

[3] See for example the Scikit-learn documentation

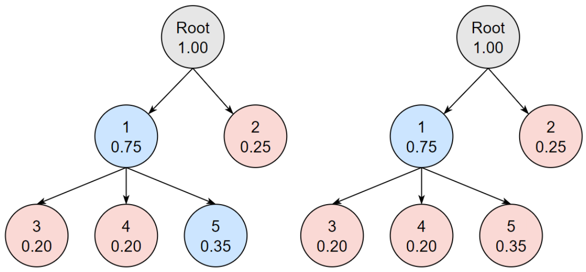

Figure 2: Example of a conditional distribution estimation over a simple hierarchy and corresponding predicted nodes (in blue) for different thresholds (0.3 on the left, 0.5 on the right).

In particular, machine learning methods often produce an estimation of $\mathbb{P}(\cdot|x)$ for a given x. If the solution of Equation (2) yields a necessary and sufficient condition on $\mathbb{P}(\cdot|x)$, this condition induces a statistically grounded inference methodology for optimizing the metric of interest. This shows how intricated the choice of inference methodology and of evaluation metric are. This statement has largely been neglected in recent HTC models, and we show in what follows that a 0.5 thresholding inference coupled with a F1-score metric can be sub-optimal.

3.2.2 On the optimality of hierarchical metrics

On Figure 2, we depict an example hierarchy as well as a coherent and exhaustive probability distribution $\mathbb{P}(\cdot|x)$ for a given x. Thresholding to 0.5 would lead to predict $\{1\}$, while we could consider prediction $\{1, 5\}$. A simple computation, detailed in Appendix B.1, gives:

$$\mathbb{E}[\text{SP}(\text{Y}, \{1\})|X = x] = 1.25$$
$$\mathbb{E}[\text{SP}(\text{Y}, \{1, 5\})|X = x] = 1.55$$
$$\mathbb{E}[\text{hF1}(\text{Y}, \{1\})|X = x] = 0.5$$
$$\mathbb{E}[\text{hF1}(\text{Y}, \{1, 5\})|X = x] = 0.55$$

This simple example shows that in a *single path leaf label* setting it is strictly better to predict $\{1\}$ instead of $\{1, 5\}$ when aiming at minimizing SP and conversely predicting $\{1, 5\}$ instead of $\{1\}$ when aiming at maximizing hF1-score. Besides the fact that optimal thresholding depends on the choice of the metric, we can show that the optimal threshold for the hF1-score depends on x (we detail the proof in Appendix B.1.2). This motivates the idea of using a per-instance hF1-score as defined in Section 3.1, rather than its *micro* version. Moreover, as the optimal threshold is unknown, we propose to evaluate hierarchical classifiers at different operating points, as proposed in Valmadre (2022). Combined with our Proposition 1, these observations motivate the re-evaluation of the current state-of-the-art models in the setting we propose in the next section.

3.2.3 Hierarchical F1-score evaluation methodology

We introduce an evaluation methodology that relies on different operating points. Broadly, this methodology involves, for a given input x, systematically exploring a range of thresholds τ. At each threshold, we calculate hPrecision and hRecall, subsequently constructing a Precision-Recall curve. More formally, let x be an input text, Y its ground truth label set, $\hat{\mathbb{P}}(\cdot|x)$, the estimated conditional distribution, and $\tau \in [0,1]$, we denote :

$$\hat{Y}^{\tau} = \{y \in \mathcal{Y}, \hat{\mathbb{P}}(y|x) > \tau\}$$

Then the Hierarchical Precision-Recall curve is defined as the set of couples

$$\left\{ \left(\mathrm{hR}(\mathrm{Y}, \hat{\mathrm{Y}}^{\tau}), \mathrm{hP}(\mathrm{Y}, \hat{\mathrm{Y}}^{\tau}) \right), \ \tau \in [0,1] \right\}$$

Curve computation. In practice, there is no need to compute values of precision and recall for all thresholds, but only for the set $\{\hat{\mathbb{P}}(y|x), y \in \mathcal{Y}\}$, as $\tau \mapsto \mathrm{hR}(\mathrm{Y}, \hat{\mathrm{Y}}^{\tau})$ and $\tau \mapsto \mathrm{hP}(\mathrm{Y}, \hat{\mathrm{Y}}^{\tau})$ are piecewise constant.

Area under the curve (AUC). After computing the hierarchical precision-recall curve, the area under this curve gives an overall performance of the estimated conditional distribution across thresholds for a given x. This is performed for each sample. AUC of all samples are then averaged across all input texts.

Now that our evaluation framework has been layed out, we will introduce our baselines, before presenting our experimental setup.

4 Simple top-down loss-based baselines

4.1 Conditional softmax cross-entropy

As outlined in Section 1, we focus on methods that, given an input text x, produce an estimated conditional distribution $\hat{\mathbb{P}}(\cdot|x)$ on $\mathcal{Y}$. We propose here to associate a modern text encoder to the conditional softmax (Redmon and Farhadi, 2017) as a strong baseline which inherently incorporates the hierarchy structure by producing a hierarchy-coherent probability distribution and coupling it with a cross-entropy loss. We detail in this section the modeling and training associated with it. Let us consider an input text x with its corresponding label set Y; a text encoder is first used to produce a embedded representation $h_x \in \mathbb{R}^d$ of x.

Conditional softmax. The conditional softmax first maps h_x to $s_x \in \mathbb{R}^{|\mathcal{Y}|}$ through a standard linear mapping:

$$s_x = W h_x + b \tag{3}$$

where $W \in \mathbb{R}^{|\mathcal{Y}| \times d}$ and $b \in \mathbb{R}^{|\mathcal{Y}|}$. Then, a softmax is applied to each brotherhood as follows:

$$\hat{\mathbb{P}}(y|x, \pi(y)) = \frac{\exp s_x^{[y]}}{\sum\limits_{z \in \mathcal{C}(\pi(y))} \exp s_x^{[z]}} \tag{4}$$

Cross-entropy. The contribution to the loss of the pair (x, Y) is given by a standard leaf nodes cross-entropy, which writes:

$$\begin{aligned} l_{\text{CSoft}}(x, Y) &= -\log \hat{\mathbb{P}}(y^{\text{spe}}|x) \\ &= -\sum_{y \in Y} \log \hat{\mathbb{P}}(y|x, \pi(y)) \qquad (5) \end{aligned}$$

where we denote y^{spe} the unique leaf node of Y.

Outputted conditional distribution. The probability of $y \in \mathcal{Y}$ is computed by a standard conditionality decomposition :

$$\hat{\mathbb{P}}(y|x) = \prod_{z \in \mathcal{A}(y)} \hat{\mathbb{P}}(z|x, \pi(z))$$

Motivations. Contrarily to BCE-based methods, this modelisation directly incorporates the hierarchy structure of labels, by definition. Besides, the outputted probability distribution is coherent and exhaustive, which fits our *single-path leaf labels* setting. It is more powerful than a leaf nodes softmax, as it decomposes the leaf probability estimation into several sub-problems. It is also computationally cheap, with a linear cost with respect to the number of nodes of $\mathcal{H}$.

Limitations. This approach involves a top-down testing phase which exposes it to data imbalance and error-propagation issues. It is also limited to the *single-path leaf labels* setting. In practice, several real-world datasets consistently used in recent literature to evaluate HTC models (Lewis et al., 2004; Aly et al., 2019) are multi-path. Also, hierarchies can be non-exhaustive, which may lead to label sets whose most specific classes are not necessarily leaf nodes. The conditional softmax is not designed for any of those cases: that is why we propose to introduce a conditional sigmoid baseline in Section 4.3.

4.2 Logit adjusted conditional softmax

Zhou et al. (2020) suggest that integrating the prior probability distribution is relevant to the HTC task, which is confirmed by their experimental results. Their approach involves initializing (or fixing) the weights of the structure encoder using this pre-computed prior distribution. We believe that the easiest way to integrate the same information into our baseline is to use a dedicated loss: we turn to the logit-adjusted softmax (Menon et al., 2021), an approach proposed to deal with class imbalance, and adapt it to the conditional softmax. Equation (4) becomes:

$$\hat{\mathbb{P}}(y|x,\pi(y)) = \frac{e^{s_x^{[y]}+\tau\log\nu(y|\pi(y))}}{\sum\limits_{z\in\mathcal{C}(\pi(y))} e^{s_x^{[z]}+\tau\log\nu(z|\pi(z))}}$$

where $\nu(y|\pi(y))$ is an estimation of $\mathbb{P}(y|\pi(y))$[4] and τ a hyperparameter. Equation (5) remains unchanged. More details on the adaptation of the logit-adjusted softmax to our case are given in Appendix B.4.2.

4.3 Conditional sigmoid binary cross-entropy

Introduced by Brust and Denzler (2020), the conditional sigmoid follows a similar intuition to conditional softmax. Sigmoids are applied to each element of s_x, modeling the conditional probability of the node given its parent. Hence, the contribution to the loss of a pair (x, Y) is given by:

$$l_{\text{CSig}}(x,Y) = -\sum_{z\in Y}\Big(\log(\hat{\mathbb{P}}(z|x,\pi(z))) + \sum_{u\in\mathcal{C}(\pi(z))\setminus\{z\}} \log\Big(1-\hat{\mathbb{P}}(u|x,\pi(z))\Big)\Big)$$

While this formula was not motivated by theoretical arguments in Brust and Denzler (2020), we can prove that gradients computed for this loss and the conditional softmax cross-entropy loss are equivalent:

$$\frac{\partial l_{\text{CSoft}}(x,Y)}{\partial W} = \frac{\partial l_{\text{CSig}}(x,Y)}{\partial W}$$

while this loss also allows to deal with both multi-path and non-exhaustive datasets. Details on gradient computation can be found in Appendix B.4.

5 Experimental settings

In this section, we introduce our datasets, models, and evaluation metrics.

[4] In practice, we estimate it by computing an empirical probability on train set for each label. It is not trainable.

5.1 Datasets

We will verify the performance of our baselines versus recent state-of-the-art models on hierarchical metrics on three widely used datasets in the HTC literature, which is mainly applied to English data: Web-of-Science (WOS) (Kowsari et al., 2018), RCV1-V2 (Lewis et al., 2004) and BGC (Aly et al., 2019). We also contribute to HTC benchmarking by releasing Hierarchical-Wikivitals (HWV), which we believe provides a harder challenge, as the number of nodes and the depth of the hierarchy are significantly higher than for the previously cited datasets. It is also characterized by a very imbalanced label distribution. We show in Figure 1 three observations from our new dataset, illustrating that leaf nodes depth can vary, ranging from 2 to 6. Table 1 shows additional data statistics. Details regarding the building process of HWV are provided in Appendix A.

Dataset	Train/Val/Test	d	#nodes	#nodes per level
HWV (SPL)	6,408/1,602 2,003	6	1186	11-109-381-437-244-4
WOS (SPL)	30,070/7,518 9,397	2	141	7-134
RCV1 (MP)	23,149/ - 781,265	4	103	4-55-43-1
BGC (MP)	58,715/14,785 18,394	4	146	7-46-77-16

Table 1: Key statistics of the selected datasets. **SPL** indicates that the dataset enters the *single path leaf labels* setting, and **MP** that it is multi-path; d represents the maximum depth of the label hierarchy.

5.2 Models

We propose to compare very different HTC models, ranging from most simple baselines to the most recent, state-of-the-art approaches. For fair comparison between them, we use a pre-trained BERT[5] model (Devlin et al., 2019) as text encoder, adopting the standard [CLS] representation as h_x for every model. We list below all the different models evaluated. **BERT + BCE** is the simplest baseline, treating the problem as a multi-label task, without using any information from the hierarchical structure of the labels. **BERT + Leaf Softmax** outputs a distribution over leaves, and hence is only fitted for single-path leaf label settings. **BERT + CHAMP** implements the penalisation of false positives based on their shortest-path distance to the ground label set in the tree (Vaswani et al., 2022). **BERT + Conditional {Softmax, logit-adjusted**

[5] https://huggingface.co/bert-base-uncased

Softmax, Sigmoid} are our strong baselines, detailed in Section 4.1. **Hitin** (Zhu et al., 2023), **HBGL** (Jiang et al., 2022), **HGCLR** (Wang et al., 2022a) are among the most recent state-of-the-art models, proposing respectively to separately encode the label hierarchy in an efficient manner, to incorporate both global and local information when encoding the label hierarchy, by considering subgraphs, and to use contrastive learning and exploiting the label hierarchy to create plausible corrupted examples. All tested methods output a conditional distribution $\hat{\mathbb{P}}(\cdot|x)$ for every input text x, except HBGL[6].

5.3 Evaluation

As shown in Section 3.2.2, given $\hat{\mathbb{P}}(\cdot|x)$, the optimal inference process depends on the chosen metric. In sections below we detail evaluation metrics depending on the setting, and the associated inference methodology.

5.3.1 Single Path datasets

Accuracy can be computed either on leaf labels, or per-level, and then averaged over levels. In both cases, we perform inference following the Bayes optimal predictor:

$$\hat{y} = \underset{y \in \mathcal{A}}{\operatorname{argmax}} \, \hat{\mathbb{P}}(y|x)$$

where $\mathcal{A}$ is the subset of nodes considered. (for leaf accuracy, $\mathcal{A} = \mathcal{L}$). For the **hF1-score**, we compute an AUC metric following Section 3.2.3. For the **shortest-path**, we follow a result from Ramaswamy et al. (2015), performing 0.5 thresholding[7] and computing the length of the shortest path between the most specific node predicted and the most specific ground truth label[8].

5.3.2 Multi Path datasets

In this setting, we replace accuracy by the **Hamming loss**, defined as:

$$\mathrm{HM}(\mathrm{Y}, \hat{\mathrm{Y}}) = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \mathbb{1}(Y_i \neq \hat{Y}_i)$$

where $\mathcal{A}$ is the set of nodes of a given level of $\mathcal{H}$. For the metric, Bayes optimal inference is performed by thresholding to 0.5 (Dembczyński et al., 2012). Then, for **hF1-score**, the methodology used for single path can be extended to multi-path without any changes.

Method	F1-score	
	Micro	Macro
BCE	**88.87 (0.15)**	45.56 (0.58)
CHAMP	87.14 (0.15)	50.90 (0.24)
HGCLR	84.92 (0.37)	44.89 (1.38)
HITIN	87.49 (0.08)	51.73 (0.42)
Leaf softmax	84.79 (0.57)	51.49 (0.52)
Cond Soft	87.20 (0.45)	53.80 (0.65)
Cond Soft (LA)	87.39 (0.21)	**54.40 (0.58)**

Table 2: F1-score (and 95% confidence interval) on HWV test set with 0.5 thresholding prediction methodology for different implemented methods. The best result is highlighted in bold. The HBGL model was too large to fit in the memory of a 40GB GPU for this dataset.

5.4 Training details

We use the `bert-base-uncased` model from the transformers library (Wolf et al., 2020) as text encoder (110M parameters). Our implementation is based on Hitin.[9] Each of our baselines is trained for 20 epochs on a V100 GPU of 32GB with a batch size of 16. We used an AdamW optimizer with initial learning rate of $2 \cdot 10^{-5}$ and with a warmup period of 10% of the training steps. For HBGL[10], Hitin and HGCLR[11], we relied on official implementations and guidelines to conduct experiments. For datasets not used in the original papers, we performed an hyperparameter optimization via grid-search. Our results are averaged over four training runs with different seeds.

6 Results and Analysis

We start our investigation by evaluating models on our newly proposed dataset, HWV, following recent literature: using 0.5 thresholding and showing *micro* and *macro* F1-score in Table 2. However, the HBGL architecture could not be run on HWV, requiring memory above the capacity of our GPUs. We additionally present in Appendix 5 these metrics for the other datasets. We can first note the remarkable efficiency of the conditional softmax on the macro-F1, especially our logit-adjusted version. Surprisingly, with a deeper, more complex hierarchy, the latest models fail to obtain the best results.

[6] This prevents us to compute the hF1-AUC metric for the HBGL model.

[7] While Ramaswamy et al. (2015) show this to be optimal in a noticeably different setting we can adapt this result to our framework.

[8] For BCE, where thresholding to 0.5 can lead to several paths, we sum the length of the shortest paths between all most specific predictions and most specific ground truth labels.

[9] https://github.com/Roooooyy/HiTIN

[10] https://github.com/kongds/HBGL

[11] https://github.com/wzh9969/contrastive-htc

Method	WOS				HWV			
	Accuracy (in %) ↑		SP ↓	hF1 AUC ↑	Accuracy (in %) ↑		SP ↓	hF1 AUC ↑
	Avg. Levels	Leaves			Avg. Levels	Leaves		
BCE	86.46 (0.10)	81.34 (0.13)	0.541 (0.003)	89.09 (0.11)	85.51 (0.20)	68.25 (0.36)	1.233 (0.028)	88.97 (0.14)
CHAMP	86.44 (0.12)	81.29 (0.12)	0.540 (0.007)	88.66 (0.09)	87.37 (0.17)	71.56 (0.30)	1.127 (0.003)	89.64 (0.20)
HBGL	**86.67 (0.12)**	**81.95 (0.13)**	**0.530 (0.006)**	×	-	-	-	-
HGCLR	86.04 (0.29)	81.02 (0.35)	0.563 (0.014)	**89.18 (0.21)**	84.58 (0.64)	67.12 (1.50)	1.211 (0.021)	88.35 (0.35)
HITIN	86.63 (0.09)	81.62 (0.08)	0.572 (0.006)	88.21 (0.06)	88.28 (0.07)	73.63 (0.22)	1.229 (0.032)	90.72 (0.16)
Leaf softmax	85.81 (0.35)	80.73 (0.51)	0.562 (0.008)	88.62 (0.08)	86.54 (0.49)	71.10 (0.72)	1.126 (0.043)	88.55 (0.47)
Cond Soft	86.04 (0.21)	80.77 (0.24)	0.546 (0.012)	88.76 (0.08)	88.51 (0.34)	73.64 (0.40)	0.953 (0.034)	90.79 (0.27)
Cond Soft + LA	85.99 (0.37)	80.62 (0.46)	0.541 (0.005)	88.90 (0.10)	**88.53 (0.25)**	**73.75 (0.25)**	**0.936 (0.016)**	**90.91 (0.11)**

Table 3: Performance evaluation metrics (and 95% confidence interval) on the test sets of the WOS and HWV datasets for the implemented models. The best result for each metric is highlighted in bold. The HBGL model was too large to fit in the memory of a 40GB GPU on the HWV dataset.

	RCV1		BGC	
Method	Hamming Loss Avg. (in %) ↓	hF1 AUC ↑	Hamming Loss Avg. (in %) ↓	hF1 AUC ↑
BCE	0.74 (0.01)	**93.59 (0.20)**	1.05 (0.03)	**90.22 (0.70)**
CHAMP	0.78 (0.04)	93.05 (0.34)	**1.03 (0.04)**	90.15 (0.22)
HBGL	**0.71 (0.01)**	×	1.06 (0.01)	×
HGCLR	0.77 (0.02)	93.09 (0.18)	1.05 (0.03)	89.65 (0.20)
HITIN	0.78 (0.03)	92.92 (0.20)	**1.03 (0.02)**	89.98 (0.14)
Cond Sigmoid	0.78 (0.07)	92.87 (0.69)	1.04 (0.02)	90.07 (0.40)

Table 4: Performance evaluation on the test sets of the RCV1 and BGC Datasets for the implemented models. The best result for each metrics are emphasized in bold. Hamming loss is displayed in %. A 95% confidence interval is also displayed.

We hence emit the hypothesis that while *global* hierarchy-aware models were proven useful on simpler datasets, they fail to capture that complexity on HWV. We then turn to hierarchical metrics to better investigate. Table 3 shows evaluation on the two *single path leaf-label* datasets: WOS and HWV. On WOS, simpler baselines reach remarkable results. Despite the marginal superiority of HBGL, it is noteworthy that the BERT+BCE model, not using label hierarchy information, is in the top performances across all metrics. This shows the low complexity of the dataset's label hierarchy. On HWV there are notable disparities: while HGCLR demonstrated low performance, and Hitin achieved average results, the conditional softmax, and the logit-adjusted version here again reach great results, and significantly outperforms other methods across nearly all metrics. We present the quantitative results for the multi path datasets in Table 4. Here, our observations align closely with what we noticed on WOS: a straightforward BCE loss consistently yields great results across datasets and metrics. As the HWV dataset is characterized by a deep hierarchy and a very imbalanced label distribution, we believe those results allow us to draw several lessons. First, that the latest state-of-the-art hierarchy-aware HTC models are in fact less able to integrate that complex hierarchical information into their prediction than a simple model trained with conditional softmax cross-entropy. Second, that it is necessary to employ appropriate data, metrics, with the right methodology, to properly evaluate a model's capacity to encode label hierarchy information.

7 Conclusion

In this paper, we come back upon recent progress in Hierarchical Text Classification, and propose to investigate closely this task's evaluation. In order to do so, we begin by showing the theoretical limitations of the inference and metrics that are commonly used in the recent literature. We instead propose to use existing hierarchical metrics, and associated inference methods, better suited for the task. Then, we propose a new and challenging dataset, Hierarchical WikiVitals; our experiments show that recent sophisticated hierarchy-aware models have trouble integrating hierarchy information, whereas simple models are very competitive. We finally propose a strong baseline, termed logit-adjusted conditional softmax cross-entropy, able to both integrate hierarchy information and deal with class imbalance on our dataset. In the future, we plan to investigate the mechanism of inference for hierarchical metrics, and will aim at making direct contribution to improving models on the HTC tasks.

Limitations

Our work emphasizes fairness and transparency, acknowledging potential limitations within the current framework. However, several key limitations remain. Firstly, our core results on metrics and inference are restricted to a specific framework we call *single-path leaf label*. Moving beyond this framework significantly increases the complexity of both evaluation and inference methodologies. Notably, in multi-path scenarios, the Shortest-path metric becomes ill-defined, necessitating consideration of often intractable label interdependencies. Secondly, we demonstrate that the commonly used 0.5 threshold is not optimal for F1-score calculation. Although we address this by considering all possible thresholds for a fair evaluation, each individual instance likely has a unique optimal threshold, which would need further research. Finally, our new incremental loss function, termed logit-adjusted conditional softmax cross-entropy is only fitted to *single-path leaf label* framework. Morever, its definition includes the computation of several cascade conditional probabilities. This means that inaccuracies in probability estimations at higher levels can disproportionately amplify errors at lower levels, potentially compromising overall model performance.

References

Rami Aly, Steffen Remus, and Chris Biemann. 2019. 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*, pages 323–330, Florence, Italy. Association for Computational Linguistics.

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, Florence, Italy. Association for Computational Linguistics.

Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A. Lord. 2020. Making better mistakes: Leveraging class hierarchies with deep networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Clemens-Alexander Brust and Joachim Denzler. 2020. Integrating domain knowledge: Using hierarchies to improve deep classifiers. In *Pattern Recognition*, pages 3–16, Cham. Springer International Publishing.

Boli Chen, Xin Huang, Lin Xiao, Zixin Cai, and Liping Jing. 2020. Hyperbolic interaction model for hierarchical multi-label classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7496–7503.

Haibin Chen, Qianli Ma, Zhenxi Lin, and Jiangyue Yan. 2021. 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)*, pages 4370–4379, Online. Association for Computational Linguistics.

Guillem Collell, Drazen Prelec, and Kaustubh Patil. 2017. Reviving threshold-moving: a simple plug-in bagging ensemble for binary and multiclass imbalanced data.

Krzysztof Dembczyński, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. 2012. On label dependence and loss minimization in multi-label classification. *Machine Learning*, 88:5–45.

Zhongfen Deng, Hao Peng, Dongxiao He, Jianxin Li, and Philip Yu. 2021. 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*, pages 3259–3265, Online. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Vivien Sainte Fare Garnot and Loic Landrieu. 2021. Leveraging class hierarchies with metric-guided prototype learning.

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*, KDD '13, page 257–265, New York, NY, USA. Association for Computing Machinery.

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.

Ting Jiang, Deqing Wang, Leilei Sun, Zhongzhi Chen, Fuzhen Zhuang, and Qinghong Yang. 2022. Exploiting global and local hierarchies for hierarchical text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4030–4039, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Svetlana Kiritchenko, Stan Matwin, Richard Nock, and A. Fazel Famili. 2006. Learning and evaluation in the presence of class hierarchies: Application to text categorization. In *Advances in Artificial Intelligence*, pages 395–406, Berlin, Heidelberg. Springer Berlin Heidelberg.

Aris Kosmopoulos, Ioannis Partalas, Eric Gaussier, Georgios Paliouras, and Ion Androutsopoulos. 2014. Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Mining and Knowledge Discovery*, 29(3):820–865.

Kamran Kowsari, Donald Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew Gerber, and Laura Barnes. 2018. Web of science dataset.

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

Volodymyr Lyubinets, Taras Boiko, and Deon Nicholas. 2018. Automated labeling of bugs and tickets using attention-based mechanisms in recurrent neural networks. In *2018 IEEE Second International Conference on Data Stream Mining and Processing (DSMP)*, pages 271–275.

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 (EMNLP-IJCNLP)*, pages 445–455, Hong Kong, China. Association for Computational Linguistics.

Aditya Menon, Harikrishna Narasimhan, Shivani Agarwal, and Sanjay Chawla. 2013. On the statistical consistency of algorithms for binary classification under class imbalance. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 603–611, Atlanta, Georgia, USA. PMLR.

Aditya Krishna Menon, Andreas Veit, Ankit Singh Rawat, Himanshu Jain, Sadeep Jayasumana, and Sanjiv Kumar. 2021. Long-tail learning via logit adjustment. In *International Conference on Learning Representations (ICLR) 2021*.

Harish Ramaswamy, Ambuj Tewari, and Shivani Agarwal. 2015. Convex calibrated surrogates for hierarchical classification. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1852–1860, Lille, France. PMLR.

Joseph Redmon and Ali Farhadi. 2017. Yolo9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium*, 6(12):e26752.

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, Brussels, Belgium. Association for Computational Linguistics.

Carlos Silla and Alex Freitas. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22:31–72.

Hyun-Je Song, Jeong-Woo Son, Tae-Gil Noh, Seong-Bae Park, and Sang-Jo Lee. 2012. A cost sensitive part-of-speech tagging: Differentiating serious errors from minor errors. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1025–1034, Jeju Island, Korea. Association for Computational Linguistics.

Junru Song, Feifei Wang, and Yang Yang. 2023. Peer-label assisted hierarchical text classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3747–3758, Toronto, Canada. Association for Computational Linguistics.

Jack Valmadre. 2022. Hierarchical classification at multiple operating points. In *Advances in Neural Information Processing Systems*, volume 35, pages 18034–18045. Curran Associates, Inc.

V.N. Vapnik. 1999. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999.

Ashwin Vaswani, Gaurav Aggarwal, Praneeth Netrapalli, and Narayan G Hegde. 2022. All mistakes are not equal: Comprehensive hierarchy aware multi-label predictions (champ).

Boyan Wang, Xuegang Hu, Peipei Li, and Philip S. Yu. 2021. Cognitive structure learning model for hierarchical multi-label text classification. *Knowledge-Based Systems*, 218:106876.

Zihan Wang, Peiyi Wang, Lianzhe Huang, Xin Sun, and Houfeng Wang. 2022a. 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)*, pages 7109–7119, Dublin, Ireland. Association for Computational Linguistics.

Zihan Wang, Peiyi Wang, Tianyu Liu, Binghuai Lin, Yunbo Cao, Zhifang Sui, and Houfeng Wang. 2022b. HPT: Hierarchy-aware prompt tuning for hierarchical text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3740–3751, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. 2018. Hierarchical multi-label classification networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5075–5084. PMLR.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Bishan Yang and Claire Cardie. 2013. Joint inference for fine-grained opinion extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1640–1649, Sofia, Bulgaria. Association for Computational Linguistics.

Alessandro Zangari, Matteo Marcuzzo, Michele Schiavinato, Matteo Rizzo, Andrea Gasparetto, Andrea Albarelli, et al. 2023. Hierarchical text classification: a review of current research. *EXPERT SYSTEMS WITH APPLICATIONS*, 224.

Xinyi Zhang, Jiahao Xu, Charlie Soh, and Lihui Chen. 2022. La-hcn: Label-based attention for hierarchical multi-label text classification neural network. *Expert Systems with Applications*, 187:115922.

Yu Zhang, Zhihong Shen, Yuxiao Dong, Kuansan Wang, and Jiawei Han. 2021. Match: Metadata-aware text classification in a large hierarchy. In *Proceedings of the Web Conference 2021*, pages 3246–3257.

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, Online. Association for Computational Linguistics.

He Zhu, Chong Zhang, Junjie Huang, Junran Wu, and Ke Xu. 2023. HiTIN: Hierarchy-aware tree isomorphism network for hierarchical text classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7809–7821, Toronto, Canada. Association for Computational Linguistics.

A Hierarchical-Wikivitals

We scraped categories of the Wikipedia 10k most-read articles[12] from a Wikipedia dump of June 2021. Each category link leads to a page with further subcategories, culminating in actual Wikipedia articles. This creates a hierarchy based on categories. For each article, we retain only its abstract as textual content and assign it all the category labels encountered while navigating the hierarchy to reach it. If inside a category we observe an actual article A and also a subgategory B, all articles inside B will be labeled the same way as A. We do that to create a leaf-label dataset. Due to inherent ambiguities in the Wikipedia category structure, the initial hierarchy formed a Directed Acyclic Graph (DAG). To enter our framework, we transformed it into a tree by differentiating categories accessed through multiple paths. This involved adding the ancestor category's name to the label for disambiguation. The resulting tree exhibits significant depth (up to 6 levels) and imbalance (leaf nodes span depths 2-6) with highly skewed label distributions (some leaf nodes have only one instance).The dataset underwent preprocessing akin to Zhou et al. (2020) to conform to standard formats and was subsequently divided into train/validation/test splits. It is available within the "data" folder of the attached repository's supplementary materials.

B Proofs

B.1 About optimal inference hierarchical metrics

B.1.1 About 0.5 thresholding

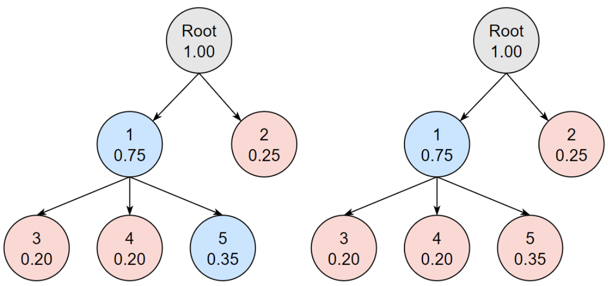

Figure 3: Example of a conditional distribution estimation over a simple hierarchy and corresponding predicted nodes (in blue) for different thresholds(*e.g.* 0.3 for left case, 0.5 for right case).

Shortest Path For the left case of Figure 3 we list all possible events and compute SP for each one.

- $\mathrm{SP}(\{1,3\},\{1,5\}) = 2$
- $\mathrm{SP}(\{1,4\},\{1,5\}) = 2$
- $\mathrm{SP}(\{1,5\},\{1,5\}) = 0$
- $\mathrm{SP}(\{2\},\{1,5\}) = 3$

Then,

$$\begin{aligned}&\mathbb{E}[\mathrm{SP}(Y,\{1,5\})|X=x]\\ =&0.2\cdot 2+0.2\cdot 2+0.25\cdot 3\\ =&1.55\end{aligned}$$

For the right case of Figure 3 we list all possible events and compute SP for each one.

- $\mathrm{SP}(\{1,3\},\{1\}) = 1$
- $\mathrm{SP}(\{1,4\},\{1\}) = 1$
- $\mathrm{SP}(\{1,5\},\{1\}) = 1$
- $\mathrm{SP}(\{2\},\{1\}) = 2$

Then,

$$\begin{aligned}&\mathbb{E}[\mathrm{SP}(Y,\{1\})|X=x]\\ =&(0.2+0.2+0.35)\cdot 1+0.25\cdot 2\\ =&1.25\end{aligned}$$

$$\mathbb{E}[\mathrm{hF1}(\mathrm{Y},\{1\})|X=x] < \mathbb{E}[\mathrm{hF1}(\mathrm{Y},\{1,5\})|X=x]$$

What we conclude from this simple computation is that it is strictly better to predict node 1 than 5 when aiming at maximizing SP.

hF1-score For the left case of Figure 3 we list all possible events and compute hF1 for each one.

- $\mathrm{hF1}(\{1,3\},\{1,5\}) = \frac{1}{2}$
- $\mathrm{hF1}(\{1,4\},\{1,5\}) = \frac{1}{2}$
- $\mathrm{hF1}(\{1,5\},\{1,5\}) = 1$
- $\mathrm{hF1}(\{2\},\{1\}) = 0$

Then,

$$\begin{aligned}&\mathbb{E}[\mathrm{hF1}(Y,\{1\})|X=x]\\ =&0.2\cdot\frac{1}{2}+0.2\cdot\frac{1}{2}+0.35\cdot 1=0.55\end{aligned}$$

For the right case of Figure 3 we list all possible events and compute hF1 for each one.

- $\mathrm{hF1}(\{1,3\},\{1\}) = \frac{2}{3}$
- $\mathrm{hF1}(\{1,4\},\{1\}) = \frac{2}{3}$

[12] https://en.wikipedia.org/wiki/Wikipedia:Vital_articles/Level/4

- $\mathrm{hF1}(\{1,5\},\{1\}) = \frac{2}{3}$
- $\mathrm{hF1}(\{2\},\{1\}) = 0$

Then,

$$\mathbb{E}[\mathrm{hF1}(\mathrm{Y},\{1\})|X=x] = 0.2 \cdot \frac{2}{3} + 0.2 \cdot \frac{2}{3} + 0.35 \cdot \frac{2}{3} = 0.5$$

$$\mathbb{E}[\mathrm{hF1}(\mathrm{Y},\{1\})|X=x] < \mathbb{E}[\mathrm{hF1}(\mathrm{Y},\{1,5\})|X=x]$$

What we conclude from this simple computation is that it is strictly better to predict node 5 when aiming at maximizing hF1. We also can conclude that optimal threshold is lower than 0.35.

B.1.2 Dependance on x of the optimal threshold

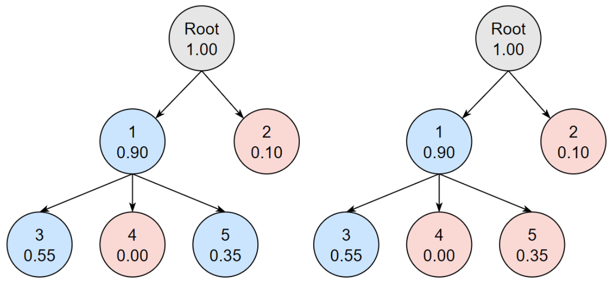

Figure 4: Example of a conditional distribution estimation over a simple hierarchy and corresponding predicted nodes (in blue) for different thresholds(*e.g.* 0.3 for left case, 0.5 for right case).

For the left case of Figure 4 we list all possible events and compute hF1 for each one.

- $\mathrm{hF1}(\{1,3\},\{1,3,5\}) = \frac{4}{5}$
- $\mathrm{hF1}(\{1,4\},\{1,3,5\}) = \frac{2}{5}$
- $\mathrm{hF1}(\{1,5\},\{1,3,5\}) = \frac{4}{5}$
- $\mathrm{hF1}(\{2\},\{1\}) = 0$

Then,

$$\mathbb{E}[\mathrm{hF1}(Y,\{1,3,5\})|X=x] = 0.55 \cdot \frac{4}{5} + 0.0 \cdot \frac{2}{5} + 0.35 \cdot \frac{4}{5} = 0.72$$

For the right case of Figure 4 we list all possible events and compute hF1 for each one.

- $\mathrm{hF1}(\{1,3\},\{1,3\}) = 1$
- $\mathrm{hF1}(\{1,4\},\{1,3\}) = \frac{1}{2}$
- $\mathrm{hF1}(\{1,5\},\{1,3\}) = \frac{1}{2}$
- $\mathrm{hF1}(\{2\},\{1\}) = 0$

Then,

$$\mathbb{E}[\mathrm{hF1}(\mathrm{Y},\{1,3\})|X=x] = 0.55 \cdot 1 + 0.0 \cdot \frac{1}{2} + 0.35 \cdot \frac{1}{2} = 0.725$$

What we conclude from this simple computation is that it is strictly better to predict node $\{1,3\}$ than $\{1,3,5\}$ when aiming at maximizing hF1. We also can conclude that optimal threshold is strictly higher 0.35 while we proved for the example of Figure 3 that the optimal threshold was below 0.35. Both examples shows that the optimal thresholds for each distribution are different and depend on x. This naturally leads us to use a *samples* hF1-score, since it makes no sense to compute a F1-score in a *micro* fashion for a given threshold for every x.

B.2 Equivalence between multilabel and hierarchical metrics

Let us consider $((Y_i, \hat{Y}_i))_{i\in[1,N]}$ of pairs of targets labels and predicted labels where

$$\forall i, \quad Y_i, \hat{Y}_i \in \{0,1\}^L$$

L is number of different categories. Let $i \in [1,N]$ and $j \in [1,L]$, we denote Y_i^j the j-th element of Y_i We define a certain number of metrics below.

B.2.1 Multi-label F1-score

We define

- The true positives of example i is the set $TP_i = \{j \in [1,L]\,,\ (Y_i^j = 1) \cap (\hat{Y}_i^j = 1)\}$
- The true negatives of example i is the set $TN_i = \{j \in [1,L]\,,\ (Y_i^j = 0) \cap (\hat{Y}_i^j = 0)\}$
- The false positives of example i is the set $FP_i = \{j \in [1,L]\,,\ (Y_i^j = 0) \cap (\hat{Y}_i^j = 1)\}$
- The false negatives of example i is the set $FN_i = \{j \in [1,L]\,,\ (Y_i^j = 1) \cap (\hat{Y}_i^j = 0)\}$

Micro F1-score

$$\mathrm{Precision_{micro}} = \frac{\sum_{i=1}^{N} |TP_i|}{\sum_{i=1}^{N} |TP_i| + |FP_i|}$$

$$\mathrm{Recall_{micro}} = \frac{\sum_{i=1}^{N} |TP_i|}{\sum_{i=1}^{N} |TP_i| + |FN_i|}$$

$$\mathrm{F_1 - score_{micro}} = \frac{2 \cdot \mathrm{Precision_{micro}} \cdot \mathrm{Recall_{micro}}}{\mathrm{Precision_{micro}} + \mathrm{Recall_{micro}}}$$

Samples F1-score

$$\mathrm{Precision_i} = \frac{|TP_i|}{|TP_i| + |FP_i|}$$

$$\mathrm{Recall_i} = \frac{|TP_i|}{|TP_i| + |FN_i|}$$

$$\mathrm{F_1 - score_i} = \frac{2 \cdot \mathrm{Precision_i} \cdot \mathrm{Recall_i}}{\mathrm{Precision_i} + \mathrm{Recall_i}}$$

$$\mathrm{F_1 - score_{samples}} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{F_1 - score_i}$$

B.2.2 Hierarchical F1-score

Micro hF1-score

$$\mathrm{hPrecision_{micro}} = \frac{\sum_{i=1}^{N} \left|\hat{Y}_i^{\text{aug}} \cap Y_i\right|}{\sum_{i=1}^{N} \left|\hat{Y}_i^{\text{aug}}\right|}$$

$$\mathrm{hRecall_{micro}} = \frac{\sum_{i=1}^{N} \left|\hat{Y}_i^{\text{aug}} \cap Y_i\right|}{\sum_{i=1}^{N} |Y_i|}$$

$$\begin{aligned}&\mathrm{hF_1 - score_{micro}}\\ &= \frac{2 \cdot \mathrm{hPrecision_{micro}} \cdot \mathrm{hRecall_{micro}}}{\mathrm{hPrecision_{micro}} + \mathrm{hRecall_{micro}}}\end{aligned}$$

Samples hF1-score

$$\mathrm{hPrecision_i} = \frac{\left|\hat{Y}_i^{\text{aug}} \cap Y_i\right|}{\left|\hat{Y}_i^{\text{aug}}\right|}$$

$$\mathrm{hRecall_i} = \frac{\left|\hat{Y}_i^{\text{aug}} \cap Y_i\right|}{|Y_i|}$$

$$\mathrm{hF_1 - score_i} = \frac{2 \cdot \mathrm{hPrecision_i} \cdot \mathrm{hRecall_i}}{\mathrm{hPrecision_i} + \mathrm{hRecall_i}}$$

$$\mathrm{hF_1 - score_{samples}} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{hF_1 - score_i}$$

Proposition 2 *In micro and samples settings, if every prediction $\hat{Y}$ is coherent then hF1 and F1 are strictly equal*

We recall that we consider here predictions that are coherent meaning $y \in \hat{Y} \implies \mathcal{A}(y) \subset \hat{Y}$. In that case $Y_i^{\text{aug}} = Y_i$. In the multi-label framework the micro-precision writes :

$$\begin{aligned}\mathrm{Precision_i} &= \frac{|TP_i|}{|TP_i| + |FP_i|}\\ &= \frac{\overbrace{\sum_{y \in \hat{Y}_i} \mathbb{1}(y \in Y_i)}^{=\left|\hat{Y}_i \cap Y_i\right|}}{\underbrace{\sum_{y \in \hat{Y}_i} \underbrace{\mathbb{1}(y \in Y_i) + \mathbb{1}(y \notin Y_i)}_{=1}}_{=\left|\hat{Y}_i\right|}}\\ &= \frac{\left|\hat{Y}_i \cap Y_i\right|}{\left|\hat{Y}_i\right|}\\ &= \mathrm{hPrecision_i}\end{aligned}$$

Similarly,

$$\begin{aligned}\mathrm{Recall_i} &= \frac{|TP_i|}{|TP_i| + |FN_i|}\\ &= \frac{\overbrace{\sum_{y \in Y_i} \mathbb{1}(y \in \hat{Y}_i)}^{=\left|\hat{Y}_i \cap Y_i\right|}}{\underbrace{\sum_{y \in Y_i} \underbrace{\mathbb{1}(y \in \hat{Y}_i) + \mathbb{1}(y \notin \hat{Y}_i)}_{=1}}_{=|Y_i|}}\\ &= \frac{\left|\hat{Y}_i \cap Y_i\right|}{|Y_i|}\\ &= \mathrm{hRecall_i}\end{aligned}$$

And naturally,

$$\begin{aligned}\mathrm{hF1 - score_i} &= \frac{2 \cdot \mathrm{hPrecision_i} \cdot \mathrm{hRecall_i}}{\mathrm{hPrecision_i} + \mathrm{hRecall_i}}\\ &= \frac{2 \cdot \mathrm{Precision_i} \cdot \mathrm{Recall_i}}{\mathrm{Precision_i} + \mathrm{Recall_i}}\\ &= \mathrm{F1 - score_i}\end{aligned}$$

This computation was performed for *samples* but holds for the *micro* framework. This proves Proposition 1

B.3 Hierarchical logit adjustement

Our motivation is twofold :

- Incorporate prior hierarchy knowledge in our loss

- Deal with label imbalance.

In *imbalanced* standard classification one typically get rid of standard accuracy metric that can be very high even if witnessing poor results on underrepresented classes. We then want to maximize macro-accuracy. It corresponds to looking for a minimizer of the per-class error rates which writes :

$$\mathrm{BER}(f) = \frac{1}{L}\sum_{y\in[L]} \mathbb{P}_{x|y}\left(y \notin \operatorname*{argmax}_{y\in[L]} f_{y'}(x)\right)$$

This can be seen as using a *balanced* class probability function $\mathbb{P}^{\mathrm{bal}}(y|x) \propto \frac{1}{L}\mathbb{P}(x|y)$.
In our case of hierarchical classification, one typically could want to minimize leaves-balanced error which would lead to minimize

$$\mathrm{BER}(f) = \frac{1}{|\mathcal{L}|}\sum_{y\in\mathcal{L}} \mathbb{P}_{x|y}\left(y \notin \operatorname*{argmax}_{y\in\mathcal{L}} f_{y'}(x)\right)$$

Let us consider $f^* \in \operatorname*{argmin}_{f:\mathcal{X}\to\mathbb{R}^{|\mathcal{L}|}} \mathrm{BER}(f)$ the *Bayes-optimal* scorer for this problem.

Then following (Menon et al., 2013; Collell et al., 2017) we have,

$$\operatorname*{argmax}_{y\in\mathcal{L}} f_y^*(x) = \operatorname*{argmax}_{y\in\mathcal{L}} \mathbb{P}^{\mathrm{bal}}(y|x) \tag{6}$$

But,

$$\mathbb{P}^{\mathrm{bal}}(y|x) = \frac{1}{L}\mathbb{P}(x|y) \underbrace{=}_{\text{Bayes formula}} \frac{1}{L} \cdot \frac{\mathbb{P}(y|x)\mathbb{P}(x)}{\mathbb{P}(y)}$$

Then, (6) becomes :

$$\begin{aligned} \operatorname*{argmax}_{y\in\mathcal{L}} f_y^*(x) &= \operatorname*{argmax}_{y\in\mathcal{L}} \frac{1}{|\mathcal{L}|} \cdot \frac{\mathbb{P}(y|x)\mathbb{P}(x)}{\mathbb{P}(y)} \\ &= \operatorname*{argmax}_{y\in\mathcal{L}} \frac{\mathbb{P}(y|x)}{\mathbb{P}(y)} \end{aligned} \tag{7}$$

Now suppose, as in the conditional softmax framework, that, for a given $y \in \mathcal{Y}$, we have $\mathbb{P}(y|x, \pi(y)) \propto \exp s_y^*(x)$ for an unknown optimal scorer $s^* : \mathcal{X} \to \mathbb{R}^{|\mathcal{Y}|}$.

Then, (7) becomes :

$$\begin{aligned} \operatorname*{argmax}_{y\in\mathcal{L}} f_y^*(x) &= \operatorname*{argmax}_{y\in\mathcal{L}} \prod_{z\in\mathcal{A}(y)} \frac{\overbrace{\mathbb{P}(z|x,\pi(z)))}^{=\exp(s_z^*(x)}}{\mathbb{P}(z|\pi(z))} \\ &= \operatorname*{argmax}_{y\in\mathcal{L}} \exp\left(\sum_{z\in\mathcal{A}(y)} s_z^*(x) - \log \mathbb{P}(z|\pi(z))\right) \\ &= \operatorname*{argmax}_{y\in\mathcal{L}} \sum_{z\in\mathcal{A}(y)} s_z^*(x) - \log \mathbb{P}(z|\pi(z)) \end{aligned} \tag{8}$$

As in Menon et al. (2021) this suggests training a model to estimate directly $\mathbb{P}^{\mathrm{bal}}$ whose logits are implicitly modified as per (8) which would yield the following loss :

$$l_{\mathrm{CSoLa}}(x, y) = -\sum_{z\in\mathcal{A}(y)} \log \hat{\mathbb{P}}(z|x, \pi(z))$$

Where

$$\hat{\mathbb{P}}(y|x, \pi(y)) = \frac{e^{s_x^{[y]} + \tau \log \nu(y|\pi(y))}}{\sum_{z\in\mathcal{C}(\pi(y))} e^{s_x^{[z]} + \tau \log \nu(z|\pi(z))}}$$

where $\nu(y|\pi(y))$ is a estimation of $\mathbb{P}(y|\pi(y))$ and τ an hyperparameter (which would be optimally 1).

B.4 Link between Conditional Softmax and conditional sigmoid

B.4.1 Conditional softmax gradient computation

We compute the gradient of the loss with respect to the final weight matrix to understand how parameters of the last layer are updated with the conditional framework. Let first express the loss in terms of the weights of the last layer.

$$\mathcal{L}_x = -\sum_{z\in\mathcal{A}(y)} \log \hat{\mathbb{P}}(z|x, \pi(z))$$

$$\mathcal{L}_x = -\sum_{z\in\mathcal{A}(y)} \log\left(\frac{\exp(W_{[z]}^T h_x + b_{[z]})}{\sum_{j\in\mathcal{C}(\pi(z))} \exp(W_j^T h_x + b_j)}\right)$$

$$= -\sum_{z\in\mathcal{A}(y)} \left(W_{[z]}^T h_x + b_{[z]} + \log\left(\sum_{j\in\mathcal{C}(\pi(z))} \exp(W_j^T h_x + b_j)\right)\right)$$

Then, we consider the set weights $\mathcal{I}_y = \{\mathcal{C}(\pi(z)),\ z \in \mathcal{A}(y)\}$. It correspond to the weights involved in the expression of $\mathcal{L}_x$.

Let $k \in [0, |\mathcal{Y}| - 1]$,

Method	WOS		HWV		RCV1		BGC	
	F1-score		F1-score		F1-score		F1-score	
	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro
BCE	87.02 (0.05)	81.19 (0.12)	**88.87 (0.15)**	45.56 (0.58)	86.65 (0.30)	66.47 (1.49)	80.12 (0.70)	60.40 (3.49)
CHAMP	87.01 (0.13)	81.23 (0.18)	87.14 (0.15)	50.90 (0.24)	85.76 (0.58)	61.63 (3.46)	80.11 (0.78)	60.98 (4.51)
HBGL	**87.22 (0.10)**	**81.86 (0.19)**	-	-	**87.01 (0.37)**	**69.52 (1.04)**	79.77 (0.13)	**64.80 (0.24)**
HGCLR	86.63 (0.28)	80.04 (0.45)	84.92 (0.37)	44.89 (1.38)	86.12 (0.26)	67.49 (0.61)	**80.16 (0.29)**	63.58 (0.40)
HITIN	87.05 (0.10)	81.49 (0.07)	87.49 (0.08)	51.73 (0.42)	85.72 (0.52)	60.00 (4.46)	80.08 (0.51)	59.90 (3.18)
Leaf softmax	85.91 (0.25)	80.02 (0.29)	84.79 (0.57)	51.49 (0.52)	–	–	–	–
Cond Soft	86.27 (0.17)	80.26 (0.34)	87.20 (0.45)	53.80 (0.65)	–	–	–	–
Cond Soft (LA)	86.35 (0.12)	80.11 (0.26)	87.39 (0.21)	**54.40 (0.58)**	–	–	–	–
Cond Sigmoid	–	–	–	–	85.97 (0.88)	65.32 (0.87)	79.59 (1.00)	61.01 (2.35)

Table 5: F1-score on the Test Set of all Datasets for Different Implemented Methods and for 0.5 thresholding methodology. Significant and Superior Metric is Emphasized in Bold. A 95% confidence interval is also displayed.

- If $k \notin \mathcal{I}_y$ then

$$\frac{\partial \mathcal{L}_x}{\partial w_k} = 0$$

- If $k \in \mathcal{I}_y$ then

$$\begin{aligned} \frac{\partial \mathcal{L}_x}{\partial w_k} &= -\mathbb{1}_{k \in \mathcal{A}(y)} h_x \\ &\quad + \underbrace{\frac{\exp(w_k^T h_x + b_k)}{\left(\sum_{j \in \mathcal{C}(\pi(k))} \exp(w_j^T h_x + b_j)\right)}}_{\hat{\mathbb{P}}(k|x,\pi(k))} h_x \\ &= -\left(\mathbb{1}_{k \in \mathcal{A}(y)} - \hat{\mathbb{P}}(k|x,\pi(k))h_x\right. \end{aligned}$$

B.4.2 Link with Conditional Sigmoid

In Section 4.3, we introduced the conditional sigmoid, we propose here to provide some justification of the loss employed. (masking BCE introduced in (Bertinetto et al., 2020))

We recall the definition:

$$\hat{\mathbb{P}}(y|x,\pi(y)) = \frac{1}{1 + \exp - s_x^{[y]}}$$

Where

$$s_x = W^T h_x + b \;\; (W \in \mathbb{R}^{d \times |\mathcal{Y}|},\; b \in \mathbb{R}^{|\mathcal{Y}|})$$

And then the contribution to the loss of the input text/label x, y is given by Cross-Entropy loss as follows :

$$= -` \sum_{z \in \mathcal{A}(y)} \left(\log(\hat{\mathbb{P}}(z|x,\pi(z))) + \sum_{u \in \mathcal{C}(\pi(z)) \setminus \{z\}} \log(1 - \hat{\mathbb{P}}(u|x,\pi(z))) \right)$$

Considering an identical approach as in Section B.4.1 we show that :

- If $k \notin \mathcal{I}_y$ then

$$\frac{\partial \mathcal{L}_x}{\partial w_k} = 0$$

- If $k \in \mathcal{I}_y$ then

$$\frac{\partial \mathcal{L}_x}{\partial w_k} = -\left(\mathbb{1}_{k \in \mathcal{A}(y)} - \hat{\mathbb{P}}(k|x,\pi(k))\right) h_x$$

Which is exactly the same updates formulas as for the Conditional Softmax. This justifies why we consider such a loss when implementing the conditional sigmoid framework.