Label Augmentation for Zero-Shot Hierarchical Text Classification

Lorenzo Paletto and **Valerio Basile** and **Roberto Esposito**
University of Turin, Computer Scince Department
Corso Svizzera 185, 10149 Torino
{lorenzo.paletto,roberto.esposito,valerio.basile}@unito.it

Abstract

Hierarchical Text Classification poses the difficult challenge of classifying documents into multiple labels organized in a hierarchy. The vast majority of works aimed to address this problem relies on supervised methods which are difficult to implement due to the scarcity of labeled data in many real world applications. This paper focuses on strict Zero-Shot Classification, the setting in which the system lacks both labeled instances and training data. We propose a novel approach that uses a Large Language Model to augment the deepest layer of the labels hierarchy in order to enhance its specificity. We achieve this by generating semantically relevant labels as children connected to the existing branches, creating a deeper taxonomy that better overlaps with the input texts. We leverage the enriched hierarchy to perform Zero-Shot Hierarchical Classification by using the Upward score Propagation technique. We test our method on four public datasets, obtaining new state-of-the art results on three of them. We introduce two cosine similarity-based metrics to quantify the density and granularity of a label taxonomy and we show a strong correlation between the metric values and the classification performance of our method on the datasets.

1 Introduction

Hierarchical Text Classification (HTC) (Sun and Lim, 2001; Stein et al., 2019) is a Machine Learning problem that consists in classifying documents into multiple labels which are organized in the form of a hierarchical taxonomy. In recent times, this problem has increasingly gathered the interest of both academia and industry due to its relevance in realistic scenarios (Meng et al., 2019). In fact, real-world challenges such as the organization of products in e-commerce categories or the classification of documents such as papers or news in a hierarchical structure can be tackled by HTC (Song and Roth, 2014).

The greatest difficulty found in this practice is the lack of labelled data and the cost, especially in an industrial framework, of manually annotating data samples. Moreover, the structure of a hierarchy can change in time and gain or lose classes, which, potentially, can result in additional costs necessary to reorganize existing data and retrain models. For these reasons, researchers have turned their attention to Few-Shot (Snell et al., 2017) and Zero-Shot Classification (ZSC) (Song and Roth, 2014) settings in which only few or no annotated documents are given at training time. In this paper we will focus on *strict* ZSC: a highly constrained scenario where not even the unlabeled training instances are given to the system.

In the context of textual classification the standard approach used to tackle a strict ZSC problem is to transform it into a textual entailment task solved by a LLM such as as BART-MNLI (Yin et al., 2019; Williams et al., 2017). In this approach, the LLM is asked to determine if a premise sentence (the text to be classified) entails semantically a hypothesis sentence (the class to be predicted). Even without fine-tuning, these models are able to classify documents into unseen classes with a high degree of success. Another approach to the ZSC task is to use labels embeddings as prototypes or centroids for a 1-Nearest Neighbor classification problem (Snell et al., 2017; Liu et al., 2023). Input texts are vectorized in the same embedding space as the labels and the corresponding class is determined via some distance or similarity metric such as the cosine similarity. While this is a natural approach, it has been criticized (Bongiovanni et al., 2023; Rondinelli et al., 2022) on the basis that it looks for similarities between few or single words and long and complex texts. Furthermore, documents with a very high level of detail may pose a challenge for models to accurately classify them. For instance, a product review categorized under "strollers" might solely discuss the instability expe-

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 7697–7706
August 11-16, 2024

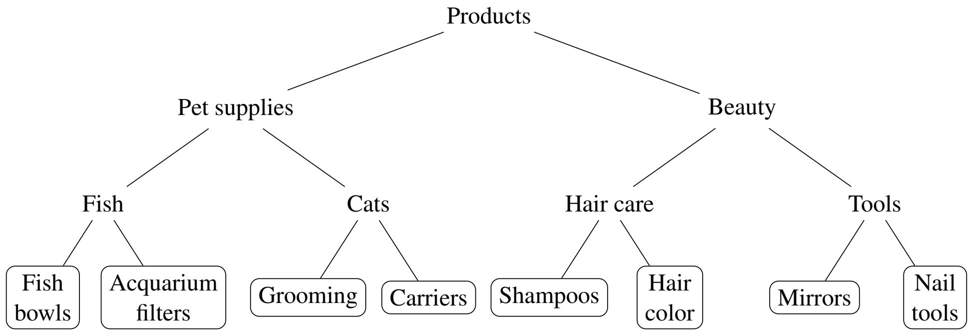

Figure 1: An example of a deepened taxonomy. Boxed labels are added by HiLA.

rienced when using the three wheels on sidewalks, making it challenging for the model to identify the appropriate label.

While there exists a growing body of research focusing on ZSC, only few works deal with hierarchical data. A recent work (Bongiovanni et al., 2023) proposes a zero shot HTC (ZS HTC) model that exploits the labels' taxonomy to improve classification results. The authors compute the similarity between the text to be classified and all the labels in the hierarchy, then they define a technique called Upward score Propagation (UP) that propagates similarity scores upward in the hierarchy and exploits the propagated information to improve the classification of the upper levels of the hierarchy. Although this technique takes advantage of the taxonomy, it cannot improve the classification results for the deepest level of labels (i.e., the leaves of the hierarchy) which, we argue, are often the most important to be classified correctly in practical applications.

This paper introduces Hierarchical Label Augmentation (HiLA), a novel technique aimed at enhancing a provided label hierarchy by leveraging a Large Language Model (LLM) to introduce meaningful branches to the existing taxonomy. Namely, we augment the deepest layer of the hierarchy by generating terms that are connected as children of the existing leaves (see Figure 1). The idea behind this process is to augment the taxonomy specificity so that the new layer of labels gets semantically closer to input texts. We then apply UP as in Bongiovanni et al. (2023) to the deepened hierarchy to perform Zero-Shot Hierarchical Classification. We test our method on four public datasets and obtain new state-of-the-art results for three of them. Moreover, we define a set of cosine similarity-based metrics to quantify the granularity of a taxonomy of labels. We conjecture that the accuracy of our approach highly depends on how granular the leaves of the taxonomy are. Indeed we show that the metrics values for the four datasets taxonomies are strongly correlated with the results of our method. Empirical results show that the metrics can be used as a prior test to measure the goodness of a label hierarchy and to check if our proposed method is going to improve the final classification results.

The main contributions of our paper are:

1. the introduction of a novel technique to augment a label hierarchy;
2. the extension of the UP technique to support improvements in the classification of the leafs of a given hierarchy;
3. definition of a metric measuring cluster density, which correlates with how well the newly proposed method works;
4. an assessment of the given technique, comparing the newly introduced method with the state-of-the-art.

The rest of the paper is organized as follows: in Section 2 we point out relevant related works. In Section 3 we summarize the UP procedure, we present our novel method and two metrics to examine taxonomy structures. We then comment the experiments we performed and their results in sections 4 and 5. Finally, we draw brief conclusions in Section 6.

2 Related Works

Many past works have studied the problem of HTC (Silla and Freitas, 2011) and have found solutions based on Machine Learning models such as Decision Trees (Vens et al., 2008) or Support Vector machines (Dekel et al., 2004). Since the advent of Transformers (Vaswani et al., 2017) more recent studies approached the problem using advanced Deep Learning (DL) techniques and Language Models (LM). In Kowsari et al. (2017) the authors take into account the hierarchical structure of the taxonomy by training a different Deep Neural Network on each node of the taxonomy. In this way, they employs stacks of deep architectures to provide specialized understanding at each level of the document hierarchy. In Huang et al. (2019), the researchers develop a DL methodology to capture both local and global information across various levels of the taxonomy. They first learn representations for both the document and the taxonomy and then employ an attention mechanism to model dependencies in a top-down manner. Finally, a classifier is used to decide whether a document merits labeling with a specific node. All the mentioned works leverage the hierarchical structure of the labels but specifically rely on labeled data.

The challenge of Zero-Shot Classification has been the focus of attention in recent years and has produced many works proposing appealing solutions. In Gera et al. (2022) the authors address the ZSC problem with a Self-Training based approach. They first compute the similarity of a document with all the labels. Secondly, they select the highest scoring documents and confidently treat them as labeled data. They use then the self-labeled documents as data to fine-tune a LM. Yin et al. (2019); Williams et al. (2017); Pàmies et al. (2023); Puri and Catanzaro (2019) propose to deal with ZSC as a textual entailment problem. They convert the labels into the hypothesis *I="This document talks about* [label]*"* and use LLMs to decide if *I* entails the document. All the cited methods either do not address ZSC in a strict sense or are not able to leverage the labels taxonomy structure.

When handling contexts with limited data it is a common practice to resort to Data Augmentation (DA) strategies (Bayer et al., 2022). Several successful techniques have been presented in this field, most of them perform DA in the feature space of vectorized texts (Kumar et al., 2019) or at text level via generative models (Edwards et al., 2021; Zhou et al., 2021). All these methods are applicable in a Few Shot context only. In the domain of Few or ZSC, it is a common strategy to employ the class taxonomy to mitigate the absence of data. This approach has been widely adopted across various fields, including Computer Vision (Fonio et al., 2023) but, to the best of our knowledge, there are no works trying to perform data augmentation over the label hierarchy.

A work strongly aligned with ours, and which we deeply rely on, is Bongiovanni et al. (2023), where ZS HTC is performed taking advantage of hierarchical information. The authors present a new methodology for text classification based on a custom hierarchical taxonomy, achieved without relying on labeled data. Their approach initially involves leveraging semantic information rooted inside pre-trained Deep Language Models to assign a preliminary relevance score to each label of the taxonomy through zero-shot techniques. Next, they leverage the hierarchical structure to reinforce the initial scores, thereby improving the overall classification process. While they do not update the relevance scores for the last level of the taxonomy, our work is strongly directed towards improving them.

3 Method

In this section, we provide an overview of the UP method as outlined in Bongiovanni et al. (2023). Subsequently, we present our proposed label augmentation method designed to enhance a given label hierarchy by adding an additional level of labels. Afterwards, we illustrate the UP application to the novel hierarchy generated by our approach. Finally, we introduce a set of metrics aimed at quantifying the granularity of label hierarchies.

We will use a notation largely inspired by Silla and Freitas (2011) and Fagni and Sebastiani (2007), extending and customizing it to align with the specific requirements of our study. Specifically, we will use an upward arrow to denote the parent of a node in the hierarchy, for instance $\uparrow c_j^l$ is the parent node of the j-th label at level l of the hierarchy. A fat upward arrow will denote the set of all ancestors of a given node (as in, e.g., $\Uparrow c_j^l$). Similarly, we will use a downward arrow to denote the set of children of a given category (e.g., $\downarrow c_j^l$) and a fat downward arrow to denote the set of all descendants categories of a given category (e.g., $\Downarrow c_j^l$). A summary of the notation is presented in Table 1 for quick reference.

3.1 ZS HTC

In this subsection we briefly summarize the Upward Score Propagation procedure introduced by Bongiovanni et al. (2023).

A document d and a label c_j^l belonging to a hier-

Symbol	Meaning
l	A level of the hierarchy, $l = 0, \ldots, L$
c_j^l	Label number j at level l
N_l	The number of labels in a level of the taxonomy
c^0	The root of the hierarchy, usually a nameless label or the name of the dataset
$\uparrow c_j^l$	The parent category of class c_j^l
$\Uparrow c_j^l$	The set of ancestor categories of class c_j^l.
$\downarrow c_j^l$	The set of children of class c_j^l
$\Downarrow c_j^l$	The set of descendant categories of class c_j^l.
$(\downarrow c_j^l)_i$	The i-th children of class c_j^l

Table 1: Notation for the class hierarchy. Since we consider tree-shaped hierarchies, $\uparrow c_j^l$ consists of one label and $\Uparrow c_j^l$ consists of one label for each level $l' < l$. Moreover, $\uparrow c_j^1 = \Uparrow c_j^1 = c^0 \ \forall j = 1, \ldots, N_1$ and $\downarrow c_j^L = \{\}$ $\forall j = 1, \ldots, N_L$.

archy H are separately mapped in the same semantic vector space with Ψ_d and Ψ_c respectively. In the vector space a *prior relevance score* is computed between two embeddings:

$$p(c_j^l) = S_C(\Psi_d(d), \Psi_c(c_j^l)),$$

where S_C is the cosine similarity. $p(c_j^l)$ is computed for a document with respect to all the labels of the taxonomy. The authors then define the UP, a method which updates prior relevance scores of labels into *posterior scores* $S_{\text{UP}}(c_j^l)$ by propagating confidence scores upwards through the taxonomy. It is based on the paradigm that if a label is relevant to a document, then also its parent is.

The UP score is defined as it follows. For labels at depth L the score is simply defined as $S_{\text{UP}}(c_j^L) = p(c_j^L)$. The score for a label c_j^l at level $l < L$ is defined recursively and it requires the introduction of few pieces of notations. Let us denote with n to be the number of children of label c_j^l (i.e., $n = |\downarrow c_j^l|$), and define the score $S_{\text{UP}}^{(i)}(c_j^l)$ in function of the i-th children of c_j^l as:

$$S_{\text{UP}}^{(i)}(c_j^l) = \begin{cases} \max(p(c_j^l), 0) & \text{if } i = 0, \\ S_{\text{UP}}^{(i-1)}(c_j^l) & \text{if } (\downarrow c_j^l)_i \prec c_j^l, \\ S_{\text{UP}}^{(i-1)}(c_j^l) \cdot e^{\delta_{j,i}} & \text{if } c_j^l \prec (\downarrow c_j^l)_i, \alpha_{c_j^l}, \\ S_{\text{UP}}\left((\downarrow c_j^l)_i\right) & \text{if } (\downarrow c_j^l)_i \succ \alpha_{c_j^l}, \end{cases} \tag{1}$$

where $c \prec c', \alpha$ iff $S_{\text{UP}}(c) < \min(S_{\text{UP}}^{(i-1)}(c'), \alpha)$ assuming $c' = \infty$ or $\alpha = \infty$ when they are not specified, and $\delta_{j,i} = S_{\text{UP}}\left((\downarrow c_j^l)_i\right) - S_{\text{UP}}^{(i-1)}(c_j^l)$.

$S_{UP}(c_j^l)$ is defined to be equal to $S_{UP}^{(n)}(c_j^l)$.

$\alpha_{c_j^l}$ represents the value above which a text is considered strongly related to the label. The second clause in the definition of the $S_{\text{UP}}^{(i)}$ function does not update its value if the i-th children of c_j^l is not relevent. The third clause updates $S_{\text{UP}}^{(i-1)}(c_j^l)$, i.e., the UP score computed up to now, multiplying it by an exponential term based on the difference in relevance between the score of the relevant children of the current node and the UP score for the current node. The last clause replaces the score of the father c_j^l entirely with the one of its son $(\downarrow c_j^l)_i$ if the score of the son is greater than $\alpha_{c_j^l}$. The final predicted label for d at level l is computed as the *argmax* of all the scores of the corresponding level.

3.2 Label augmentation

In this subsection we describe how HiLA works and describe how it is applied to a deepened hierarchy.

We assume to be given a dataset D whose labels are arranged in a hierarchy H of depth L. We propose to use a pre-trained LLM to deepen the class hierarchy by adding to every existing leaf c_j^L a set of new leaves $\downarrow c_j^L$ so that they are coherent with the original hierarchy and more specific than c_j^L. We assume that nodes of H have one and only one parent, so that the hierarchy can be represented as a tree.

We prompt an LLM to generate $\downarrow c_j^L$ starting from a context that we extract from the hierarchy itself. In principle, we would like to to include the full hierarchy in the prompt and to ask the LLM to produce a set of $\downarrow c_j^l$ for all the N_L leaves of the taxonomy. Unfortunately, such approach is intractable in many cases for two reasons: *i*) the prompt may not fit in the limited number of tokens that can be digested in a single step by the LLM[1]; *ii*) the output would itself be too long or complex to be reliably produced by the LLM.

For these reasons we propose an iterative approach where the extended hierarchy for all children of a node c_j^{L-1} at level $L-1$ are generated simultaneously and independently from the other nodes. We define the branch-set B_j^{L-1} of a node c_j^{L-1} as the set containing all labels in the branch containing c_j^{L-1} (including c_j^{L-1} itself) along with all the children of c_j^{L-1}. Formally:

$$B_j^{L-1} = c_j^{L-1} \cup \Uparrow c_j^{L-1} \cup \downarrow c_j^{L-1}. \tag{2}$$

We note that hierarchies with depth $L \geq 2$ generate branch-sets that share at least one label (c^0) and up to $L-2$ labels, if the two nodes at level $L-1$ share their father. A graphical representation of branch-sets is provided by Figure 2. Given a branch-set B_j^{L-1}, we compose the general prompt structure in the following way:

$[c_j^{L-1}][\Uparrow c_j^{L-1}]$`objects can be classified as` $[\downarrow c_j^{L-1}]$`, could you give me some more specific classifications for these classes?` (3)

where square brackets are used to denote places where the box template is filled with contextual values. The actual prompt structure can depend on the dataset on which the method is used. For the sake of exemplification, let us consider a dataset of product reviews, where at the $L-1$ level we have a label “skin care” with ancestor “beauty” and children “face”, “body” and “sun”. The branch-set of “skin care” would then be formed by “skin care”, “beauty”, “face”, “body”, and “sun”; the prompt for deepening the structure rooted in “skin care” would be:

```
"skin care" "beauty" "products" can be
classified as "face", "body" or "sun",
could you give me some more specific
classifications for these classes?
```

[1]While the biggest models are nowadays able to deal with tens of kilo-tokens, smaller models are still limited in the number of tokens they are able to digest in a single step.

We find the use of branch-sets defined in equation 2 to be more convenient than simple branches, i.e. the sets $c_j^L \cup \Uparrow c_j^L$, for two reasons: *i*) HiLA requires less LLM calls which imply less waiting/overall time; *ii*) they ensure that newly generated labels do not overlap neither with existing labels nor previously generated ones. In the example illustrated above, the generation starting from the branch defined by labels "skin care", "beauty" and "body" could create the label "face and body lenitive oils", that would overlap with an existing label of the taxonomy. Further generation hyperparameters can be specified in the prompt to meet more stringent requirements. For instance, the levels of formality, verbosity of the new labels and either a maximum or minimum number of children $|\downarrow c_j^L|$ can be added to the prompt as required.

If the hierarchy H is deepened with our label augmentation method, we can apply UP as described in equation 1. This time the labels belonging to level L of the input taxonomy are updated too because they have children. The update only happens if at least one of the generated labels is more relevant to the text than its parent label, i.e., if the label augmentation technique was effective. It is worth noting that the labels generated by our approach are not meant as classification targets for the downstream ZSC task. They just provide more context to the classification step, thus allowing for better predictions. Pseudo code for the label augmentation algorithm is shown in Algorithm 1. The algorithm calls two helper functions: the “fill_template” function uses the template given in Eq. 3 to populate the objects in the input branch-set; the “parse” function analyzes the LLM output and retrieves the set of generated labels.

Algorithm 1: The HiLA algorithm

Data: Labels hierarchy H
Result: H is extended with a new leaf level.
for $j \in N_{L-1}$ **do**
  $B_j^{L-1} \leftarrow c_j^{L-1} \cup \Uparrow c_j^{L-1} \cup \downarrow c_j^{L-1}$
  $P \leftarrow$ fill_template(B_j^{L-1}) ▷ see (3)
  $\downarrow c_j^L \leftarrow$ parse(LLM(P))
end

3.3 Cluster density estimation

Our proposed method heavily depends on the quality of the structure represented in the hierarchy. Given that the generation of new labels relies on

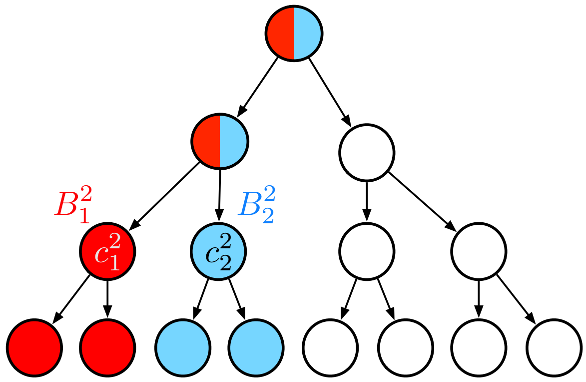

Figure 2: Examples of two branch-sets built on the c_1^2 (red) node and on the c_2^2 node (blue). Nodes shown as half red and half blue are shared among the two branch-sets.

the existing ones as prompts, the density of label embeddings becomes a critical determinant of the generated labels' quality. A higher density implies a more semantically rich and well-organized label space, thereby enhancing the efficacy of the deepening process. In this subsection we present two metrics grounded in cosine similarity to gauge the evolving average proximity among nodes as we navigate through different levels of the taxonomy. We will show in Section 5 that D_1 measure correlates with the quality of the proposed solution and measure D_2 allows for a better understanding of the hierarchy structure that will prove helpful when we will analyze the behaviour of HiLA.

We define a *label cluster* (or simply a *cluster*) to be the set:

$$C_j^{l+1} = c_j^l \cup \downarrow c_j^l, \tag{4}$$

i.e. a label and its children. The metric D_1 is defined as

$$D_1(C_j^{l+1}) = \frac{\sum_{i,k} S_C(\Psi_c((\downarrow c_j^l)_i), \Psi_c((\downarrow c_j^l)_k))}{\binom{|\downarrow c_j^l|}{2}} \tag{5}$$

for $i = 1, \ldots, |\downarrow c_j^l|, \quad k < i$, i.e. the average cosine similarity among the node's children embeddings. It measures how much the children of a node are close to each other. Metric D_2 is defined as

$$D_2(C_j^{l+1}) = \frac{\sum_{i=1}^{|\downarrow c_j^l|} S_C(\Psi_c(c_j^l), \Psi_c((\downarrow c_j^l)_i))}{|\downarrow c_j^l|}, \tag{6}$$

i.e. the average similarity between a parent node and its children. It measures how close a node is on average to its children. Upon the application of the metrics, individual D_1 and D_2 values are derived for each internal node. We summarize metric values for levels $l' : l > 0$ by averaging them across all nodes belonging to that level, i.e., we take the average of the metric values across all clusters $C_j^{l'}$ situated at the depth l'. Please note that metric D_2 has no value for $l' = 1$ since c^0 lacks an embedding representation.

4 Experiments

4.1 Data

To test the validity and versatility of our method, we select four HTC datasets with diverse content, style, and taxonomy depth. All datasets contain English text.

DBPedia Classes[2] - This dataset consists of about 340,000 Wikipedia articles that are categorized according to DBpedia's hierarchy of classes. The dataset covers different kinds of entities such as persons, places, organizations, and abstract concepts. The taxonomy has three levels with 9, 70 and 219 classes respectively. The language used is clear and refined.

Web Of Science[3] - This dataset contains about 46,000 abstracts of research papers from various scientific domains, extracted and annotated in Kowsari et al. (2017). Its taxonomy has two levels with 7 and 134 classes respectively. The language is highly technical and scientific.

Amazon Product Reviews[4] - This dataset features products reviews that are labelled according to a hierarchical taxonomy provided by Amazon. The dataset has about 50,000 reviews and its hier-

[2]https://www.kaggle.com/datasets/danofer/DBpedia-classes

[3]https://huggingface.co/datasets/web_of_science

[4]https://www.kaggle.com/datasets/kashnitsky/hierarchical-text-classification

$c_i^{L-1} \cup \Uparrow c_j^{L-1}$	c_i^L	Generated labels
Grocery gourmet food, meat & poultry	Sauces	Barbeque sauce, Soy sauce, hot sauce, pasta sauce, marinara sauce
Health and personal care, personal care	Oral hygiene	Toothpaste, Mouthwash, Toothbrushes, Tongue cleaners, Dental floss
Pet supplies, Dogs	Beds & furniture	Dog beds, Couches, Dog crates, Elevated beds

Table 2: Some examples of label generation for the Amazon dataset

Model	WoS		DBpedia			Amazon			Books	
	L. 1	L. 2	L. 1	L. 2	L. 3	L. 1	L. 2	L. 3	L. 1	L. 2
M	0.596	**0.462**	0.317	0.326	0.628	0.547	0.256	0.173	0.429	0.331
M + UP	**0.741**	**0.462**	0.759	0.656	0.628	0.712	0.348	0.173	0.566	0.331
M+HiLA +UP	0.647	0.371	**0.768**	**0.660**	**0.629**	**0.762**	**0.393**	**0.249**	**0.578**	**0.366**

Table 3: Classification F1 scores of HiLA plus UP, compared to "raw" UP and ZS HTC without the use of UP. M refers to the MPnet-based text vectorization defined in Bongiovanni et al. (2023)

archy consists of three levels with 6, 64 and 510 categories respectively. The language varies from review to review but it is typically casual and spontaneous.

Books Blurbs[5] - A set of book blurbs. The dataset taxonomy depth varied across samples, so we only used the first two levels that applied to all documents. Furthermore, we removed the texts that had more than one label per layer. After preprocessing, the dataset has about 9,000 texts and its hierarchy consists of two levels with 3 and 29 classes respectively. The tone is the one used in advertisements, the language is typically polished.

4.2 Implementation details

To perform label augmentation we rely on OpenAI API as it gives access to several LLMs of the GPT family. Specifically, we choose the `gpt-3.5-turbo` model. It is one of the best models provided by the API and it is the base version of the GPT model used by ChatGPT web interface.

For the ZS HTC and the clustering part we follow Bongiovanni et al. (2023) and use as embedder `mpnet-all` (Reimers and Gurevych, 2019; Song et al., 2020) from HuggingFace Sentence Transformers library. Results are measured in terms of macro-F1 score. All experiments are performed on a single Tesla T4 GPU[6].

5 Results

5.1 Label augmentation results

In our experiments, we did not specify a target number of generated labels, but the LLM always produced at least three labels. Some samples of the generated labels are displayed in Table 2. It is worth noting that some texts belonging to the Amazon hierarchy have some of the leaf nodes labelled as "unknown", apparently because the annotators could not associate them to any label of the taxonomy. We verified that for these labels HiLA produces new labels which are too generic and similar to the other existing labels of level L. As we will comment below, the quality of the input taxonomy is very important to the performances of the proposed approach.

The four deepened hierarchies are used to perform HTC via the UP method we introduced in Section 3.1, results are displayed in Table 3. F1-macro is chosen as classification metric following Bongiovanni et al. (2023). Applying label augmentation before UP increases results in terms of F1-score for three out of four datasets on which we

[5] https://www.inf.uni-hamburg.de/en/inst/ab/lt/resources/data/blurb-genre-collection.html

[6] Python scripts to replicate the experiments can be found at https://github.com/LPaletto/HiLA.

	Wos		DBpedia			Amazon			Books	
	L. 1	L. 2	L. 1	L. 2	L. 3	L. 1	L. 2	L. 3	L. 1	L. 2
D1	0.378	0.275	0.238	0.399	0.482	0.307	0.351	0.369	0.308	0.451
D2	-	0.393	-	0.409	0.579	-	0.436	0.444	-	0.456

Table 4: Results of metrics D_1 and D_2 applied to levels $L = l + 1$ of the hierarchical taxonomies.

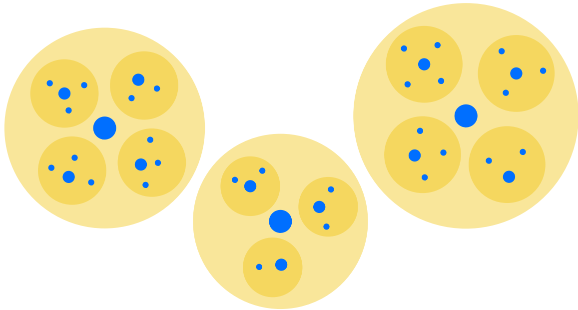

Figure 3: Visual interpretation of the positioning of the labels embeddings in a taxonomy. Blue dots represent labels embeddings, bigger blue dots symbolize higher labels in the taxonomy. Yellow areas represent space portions occupied by clusters.

achieve new state-of-the-art results. F1 increments are visible not only at the deepest level of the hierarchies L but also at all higher levels. The labels generated by our method are coherent not only with their parent labels c_j^L but also with the full branch $c_j^L \cup \Uparrow c_j^L$ to which they belong. The only dataset which does not benefit from our label augmentation technique is Web Of Science for which F1 scores worsen at every level.

5.2 Clustering density estimation results

We conducted an analysis utilizing metrics D_1 and D_2 to measure the density of each level within the hierarchical structures. The outcomes are presented in Table 4.

Upon scrutiny of the D_1 and D_2 values, a discernible trend manifests: offspring nodes exhibit a closer affinity to their parent node than to each other. This observation suggests that clusters C_j^{l+1} adopt a spatial arrangement reminiscent of an neuronal configuration, where the parent node occupies a central locus, while its progeny nodes are dispersed in the periphery. A noteworthy positive correlation between hierarchy levels and D_2 values is observed, signifying that, as we descend the taxonomy, sets $\downarrow c_j^l$ tend to concentrate more. This outcome intuitively corresponds to a progressive refinement in semantic specificity as l increases. We contend that the observed structure aligns with the expectations of a conventional hierarchy, where each branching introduces new and more refined labels to the existing taxonomy. An intuitive visual representation of the observed hierarchical spatial structure is given in Figure 3.

Contrary to the other hierarchies, Web Of Science taxonomy's semantic similarity within children of the same level decreases as the level grows. Labels of the second level move away from the existing labels instead of adding specificity to them. This results in labels becoming more distant from each other, leading to poorly defined clusters.

The analysis illustrates an evident correlation between the metrics results and the augmentation and classification results. When the taxonomy given as input to HiLA is solid in density terms, our method generates coherent labels that improve classification results once UP is applied. In contrast, if the input taxonomy lacks solidity, the generated labels fail to confer additional specificity; rather, they exacerbate the taxonomic structure, thereby deteriorating the results of UP. The analysis also confirms the hypothesis that the metrics D_1 and D_2 can be used as an initial screening tool to measure both the quality of the label taxonomy and the effectiveness of the HiLA.

6 Conclusions

In this paper we proposed an LLM-based label augmentation technique to deepen a given hierarchy of labels. We also defined a set of metrics to measure the granularity of a taxonomy. By applying our method to four public hierarchical datasets, we obtained new sets of coherent and meaningful labels. We then used the deepened hierarchies to perform Zero-Shot Hierarchical TC using the UP technique. We obtained SOTA results on three out of four datasets. The classification results are strongly correlated with the metric values, that can therefore be used to study the behaviour of the HiLA approach.

7 Limitations

The hierarchical label augmentation method we introduced depends heavily on the quality of the input taxonomy. While we tried to provide tools to assess the quality of the hierarchy, we acknowledge that the proposed approach will not provide the desired results in case the provided hierarchy is not easily extensible by the LLM. Also, our method depends on the OpenAI API, which is a closed-source tool released by a private company. To these regards, we believe that the provided approach would perform well also when coupled with models that are open source and that can be used freely (such as LLaMA (Touvron et al., 2023) or Orca-2 (Mitra et al., 2023)). Providing evidence for this claim is left as future work.

8 Acknowledgements

This research received funding from FIDA s.r.l. and from the HPC4AI project (Marco et al., 2018).

References

Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2022. A survey on data augmentation for text classification. *ACM Comput. Surv.*, 55(7).

Lorenzo Bongiovanni, Luca Bruno, Fabrizio Dominici, and Giuseppe Rizzo. 2023. Zero-shot taxonomy mapping for document classification. In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, SAC '23, page 911–918, New York, NY, USA. Association for Computing Machinery.

Ofer Dekel, Joseph Keshet, and Yoram Singer. 2004. Large margin hierarchical classification. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 27, New York, NY, USA. Association for Computing Machinery.

Aleksandra Edwards, Asahi Ushio, José Camacho-Collados, Hélène de Ribaupierre, and Alun D. Preece. 2021. Guiding generative language models for data augmentation in few-shot text classification. *CoRR*, abs/2111.09064.

Tiziano Fagni and Fabrizio Sebastiani. 2007. On the selection of negative examples for hierarchical text categorization. In *Proceedings of the 3rd language technology conference*, pages 24–28.

Samuele Fonio, Lorenzo Paletto, Mattia Cerrato, Dino Ienco, Roberto Esposito, et al. 2023. Hierarchical priors for hyperspherical prototypical networks. In *ESANN 2023-Proceedings*, pages 459–464. ESANN.

Ariel Gera, Alon Halfon, Eyal Shnarch, Yotam Perlitz, Liat Ein-Dor, and Noam Slonim. 2022. Zero-shot text classification with self-training. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1119, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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*, CIKM '19, page 1051–1060, New York, NY, USA. Association for Computing Machinery.

Kamran Kowsari, Donald E Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S Gerber, and Laura E Barnes. 2017. Hdltex: Hierarchical deep learning for text classification. In *2017 16th IEEE international conference on machine learning and applications (ICMLA)*, pages 364–371. IEEE.

Varun Kumar, Hadrien Glaude, Cyprien de Lichy, and William Campbell. 2019. A closer look at feature space data augmentation for few-shot intent classification. *CoRR*, abs/1910.04176.

Wenfu Liu, Jianmin Pang, Nan Li, Feng Yue, and Guangming Liu. 2023. Few-shot short-text classification with language representations and centroid similarity. *Applied Intelligence*, 53(7):8061–8072.

Aldinucci Marco, Rabellino Sergio, Pironti Marco, Spiga Filippo, Viviani Paolo, and et al. Maurizio, Drocco. 2018. HPC4AI, an AI-on-demand federated platform endeavour. In *ACM Computing Frontiers*, Ischia, Italy.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2019. Weakly-supervised hierarchical text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6826–6833.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes Ribeiro, Sahaj Agrawal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, Hamid Palangi, Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Awadallah. 2023. Orca-2: Teaching small language models how to reason. arXiv.

Marc Pàmies, Joan Llop, Francesco Multari, Nicolau Duran-Silva, César Parra-Rojas, Aitor Gonzalez-Agirre, Francesco Alessandro Massucci, and Marta

Villegas. 2023. A weakly supervised textual entailment approach to zero-shot text classification. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 286–296, Dubrovnik, Croatia. Association for Computational Linguistics.

Raul Puri and Bryan Catanzaro. 2019. Zero-shot text classification with generative language models. *arXiv preprint arXiv:1912.10165*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Andrea Rondinelli, Lorenzo Bongiovanni, and Valerio Basile. 2022. Zero-shot topic labeling for hazard classification. *Information*, 13(10).

Carlos N Silla and Alex A Freitas. 2011. A survey of hierarchical classification across different application domains. *Data mining and knowledge discovery*, 22:31–72.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *CoRR*, abs/2004.09297.

Yangqiu Song and Dan Roth. 2014. On dataless hierarchical text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28(1).

Roger Alan Stein, Patricia A. Jaques, and João Francisco Valiati. 2019. An analysis of hierarchical text classification using word embeddings. *Information Sciences*, 471:216–232.

Aixin Sun and Ee-Peng Lim. 2001. Hierarchical text classification and evaluation. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 521–528.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Celine Vens, Jan Struyf, Leander Schietgat, Sašo Džeroski, and Hendrik Blockeel. 2008. Decision trees for hierarchical multi-label classification. *Machine learning*, 73:185–214.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *CoRR*, abs/1704.05426.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161*.

Jing Zhou, Yanan Zheng, Jie Tang, Jian Li, and Zhilin Yang. 2021. Flipda: Effective and robust data augmentation for few-shot learning. *CoRR*, abs/2108.06332.