

HTC-GEN: A Generative LLM-Based Approach to Handle Data Scarcity in Hierarchical Text Classification

Carmelo Fabio Longo¹^a, Misael Mongiovi^{1,2}^b, Luana Bulla^{1,2}^c and Giusy Giulia Tuccari^{1,2}^d

¹National Research Council, Institute of Cognitive Science and Technology, Italy

²Department of Mathematics and Computer Science, University of Catania, Italy

Keywords: Hierarchical Text Classification, Synthetic Data Generation, Large Language Models.


Abstract: Hierarchical text classification is a challenging task, in particular when complex taxonomies, characterized by multi-level labeling structures, need to be handled. A critical aspect of the task lies in the scarcity of labeled data capable of representing the entire spectrum of taxonomy labels. To address this, we propose HTC-GEN, a novel framework that leverages on synthetic data generation by means of large language models, with a specific focus on LLama2. LLama2 generates coherent, contextually relevant text samples across hierarchical levels, faithfully emulating the intricate patterns of real-world text data. HTC-GEN obviates the need for labor-intensive human annotation required to build data for training supervised models. The proposed methodology effectively handles the common issue of imbalanced datasets, enabling robust generalization for labels with minimal or missing real-world data. We test our approach on a widely recognized benchmark dataset for hierarchical zero-shot text classification, demonstrating superior performance compared to the state-of-the-art zero-shot model. Our findings underscore the significant potential of synthetic-data-driven solutions to effectively address the intricate challenges of hierarchical text classification.


1 INTRODUCTION


Hierarchical Text Classification (HTC) is a daunting challenge in the field of Natural Language Processing, particularly when dealing with complex taxonomies. The paucity of annotated data that accurately represents the diverse spectrum of taxonomic labels has historically been a major obstacle in this field. Traditional HTC methods have often relied on labor-intensive manual annotation, potentially resulting in datasets that inadequately represent the full range of relevant taxonomic labels. We introduce Generative Hierarchical Text Classification (HTC-GEN), an automated framework that blends the strengths of Large Language Models (LLMs) with supervised approaches, dramatically reducing the need for human intervention in the HTC process. HTC-GEN's core innovation hinges on its ability to harness the generative potential of LLMs and incorporate them with state-of-the-art HTC-supervised systems in a cost-


effective manner. By significantly reducing the need for human effort, our method provides an efficient and cost-effective solution for data classification, a particularly crucial development in research areas where meticulously curated datasets are scarce, laborious to construct, or rapidly obsolete.

Our methodology focuses on creating contextually relevant text samples across all hierarchical levels using LLMs, meticulously mirroring the complexities of real-world data. Common synthetic data generation approaches involve asking LLMs to generate data for each label of the hierarchy, but this does not provide enough variability. The basic idea of our approach is to synthetically increase the granularity of the taxonomy by producing “virtual leaves”, thus to generate synthetic data on the basis of such new *extended* taxonomy. Afterward, we use the synthetic data as input to a supervision process, leveraging on state-of-the-art supervised models in the HTC field. Generating a synthetic dataset results in a notable reduction in the reliance on labor-intensive human annotation, making the process highly automated and cost-effective. This approach also aims to address the problem of imbalanced datasets in HTC, ensuring robust generalization capabilities even for labels with

^a <https://orcid.org/0000-0002-2536-8659>

^b <https://orcid.org/0000-0003-0528-5490>

^c <https://orcid.org/0000-0003-1165-853X>

^d <https://orcid.org/0009-0008-3298-7168>

minimal representation of real-world data. What further characterizes HTC-GEN is its modular structure, which allows for adaptability and alignment with future advances. The supervision module can be easily replaced with the latest and highest-performing supervised systems for HTC. This feature ensures the versatility of our methodology, aligning it with evolving technological advances.

We test our method on the Web Of Science (WoS) dataset (Kowsari et al., 2017), a corpus of nearly 50,000 scientific abstracts, each representing a specific research field. In the rapidly evolving landscape of scientific literature, efficient and cost-effective solutions for organizing and classifying research articles are in high demand. In this domain HTC plays a critical role in organizing and accessing scientific abstracts, allowing researchers to quickly locate relevant articles. However, creating accurate HTC models often requires labor-intensive manual annotations, making them potentially obsolete as new search surfaces.

Our experimental framework offers a comprehensive analysis, offering valuable insights into crucial aspects of our methodology.

We compare HTC-GEN to the state-of-the-art zero-shot model for HTC, a prevalent approach in tasks that typically entail significant human annotation efforts.

HTC-GEN, enhanced by its supervision module, achieves superior performance compared to zero-shot systems. These results are due to the extraordinary ability of generative LLMs to mimic real data and the effectiveness of HTC-based supervised approaches in learning from abundant data. Next, we test how state-of-the-art HTC systems respond to synthetic and real data inputs. The analysis involves systems trained on both synthetic and real data in a cross-case scenario. Finally, we conduct a comprehensive performance evaluation of our model across various training set sizes, offering valuable insights into its scalability and adaptability.

The main contribution of this paper can be summarized as follows:

- We introduce a novel approach that leverages on LLMs for cost-effective data generation for HTC to address the challenge of imbalanced datasets, enhance robustness and provide generalization capabilities.
- We compare our system with state-of-the-art zero-shot models and demonstrate superior performance and cost-effectiveness.
- We analyze the response of state-of-the-art HTC systems to synthetic and real data inputs, providing valuable insights for practical applications.

The paper is organized as follows. Section 2 provides an overview of the current state of the art in this field. In Section 3, we delve into the methodologies we employ, with a specific focus on data generation and supervised learning. Section 4.1 offers a comprehensive exploration of our experimental settings and results. We provide details regarding the generation of the synthetic dataset of scientific articles, analyze and compare our method with the state-of-the-art zero-shot model, and dissect the functionalities of the individual components of our system. Finally, Section 7 concludes the paper and contemplates potential future developments of our approach.

2 RELATED WORK

There are notable works addressing the problem of text categorization suitable for low-cost applications, particularly focusing on unsupervised training. Notable among these are (Haj-Yahia et al., 2019) and (Ko and Seo, 2000). Haj-Yahia et al. (Haj-Yahia et al., 2019) frame the problem as a similarity measure between documents projected onto a latent space using Latent Semantic Analysis (LSA). This method incorporates keywords assigned to each category through WordNet and expert human annotation. Similarly, Ko et al. (Ko and Seo, 2000) involve generating training sentences automatically through manually selected keywords for each category and subsequently training a Naive Bayes model for text classification. However, both approaches rely on the labor-intensive manual definition of keywords, which introduces human intervention and doesn't fully harness the semantic potential inherent in the taxonomy labels. Other works, such as (Stammbach and Ash, 2021), leverage the SBERT deep language model (Reimers and Gurevych, 2019) to encode text documents into a semantic space and create weakly supervised training sets by considering the five nearest neighbors for each data point. Nevertheless, even in this scenario, the semantic information pertaining to categories isn't fully used during the clustering process or in model training. Furthermore, they don't explicitly address the challenges of HTC.

Among the works addressing HTC (Liu et al., 2023; Zangari et al., 2023), Bongiovanni et al. (Bongiovanni et al., 2023), Bhambhoria et al. (Bhambhoria et al., 2023) and Zhang et al. (Zhang et al., 2024) stand out as the most closely aligned with our research. As our work, Bhambhoria et al. (Bhambhoria et al., 2023) focus on addressing the limitations of LLMs in various real-world scenarios, such as when the model lacks sufficient examples, faces

token vocabulary issues, or deals with a large number of label distractors. To overcome these challenges, the proposed approach combines entailment-contradiction prediction through Natural Language Inference (NLI) models with LLMs, enhancing their performance in strict zero-shot settings without requiring resource-intensive parameter updates. The study also introduces a template-based method that effectively leverages entailment and contradiction relations to improve LLM classification. The results showcase enhanced performance on hierarchical prediction tasks and the practical applicability of the approach. Although this work conducts zero-shot classification with LLM support in a pipeline methodology, the computational effort for using LLM in processing the results deviates from our proposed approach. Our method prioritizes the efficient use of LLMs capabilities during the generation of synthetic data rather than predicting and generating outcomes. As a result, we opt for a system with fewer parameters, improving versatility and cost-effectiveness. For this reason, we choose not to directly compare our methodology. Bongiovanni et al. (Bongiovanni et al., 2023) introduce a self-contained approach for HTC with a specific hierarchical taxonomy, eliminating the necessity for additional information or manual annotation. First, the author performs Zero-shot Semantic Text Classification (Z-STC) using deep language models, which generate initial relevance scores for each label in the taxonomy. During this phase, the hierarchical structure of the taxonomy is temporarily ignored, resembling a standard zero-shot text classification task. These thresholds derive from the statistical distribution of previous Z-STC scores on a set of randomly crawled Wikipedia documents, and indicate the minimum relevance score that a label must reach to be confidently considered a correct label for a given document. Finally, they initiate a score propagation process. In this phase, by propagating the relevance scores from the lowest level of the taxonomy upwards, taking advantage of both the previous Z-STC scores and the relevance thresholds, the hierarchical structure of the taxonomy is effectively incorporated. This approach not only eliminates the need for annotated data but also demonstrates significantly improved performance compared to conventional zero-shot methods.

Similar to this approach, our methodology fully adopts automation and effectively exploits the semantic aspects of the original taxonomy. However, it stands out by using this semantic information to generate synthetic data. The generated data is then used in a supervision phase that optimizes performance within a specific domain. Furthermore, our method

exploits the automated data generation facilitated by LLMs in a computationally efficient manner.

Zhang et al. (Zhang et al., 2024) propose TELE-Class, a comprehensive framework for weakly supervised HTC, which alleviate human effort for annotating training data. They employ LLMs for annotating the corpus, use the annotated corpus to extend the taxonomy, augment the initial corpus to increase class coverage and train a classification model with the enriched corpus. They test their data on Amazon-531 and DBPedia-298 datasets, showing superiority w.r.t other weakly supervised models and zero-shot models on these datasets. However their method requires to have an initial corpus of documents to start with. In contrast, our approach do not require a real dataset of documents since all documents are synthetically generated, therefore it is directly comparable to zero-shot models. For this reason, our method and the Zhang et al. approach are not directly comparable.

Finally, our approach shares commonalities with a research direction that focuses on the generation of synthetic data using LLMs (Liu et al., 2023). In (Jeronymo et al., 2023), the authors introduce InPairs-v2, conceived as a dataset generator employing open-source language models and robust rankers to generate synthetic query-document pairs. This innovation results in notable performance enhancements when compared to proprietary LLM-based models like GPT-3 (Brown et al., 2020) and FLAN (Wei et al., 2021). However, it's important to note that while InPairs-v2 targets a retrieval problem, our approach focuses on an HTC setting.

3 METHODOLOGY

The employed methodology involves the generation of a synthetic dataset by means of a LLM (LLaMa-2), as a preprocessing step (Fig. 1). The generation process takes as input a taxonomy of labels and generates synthetic data based on such taxonomy. After such pre-processing step, the synthetic dataset is given as input to the Hierarchy Guided Contrastive Learning (Wang et al., 2022) (HGCLR) base module for training, which produces the final HGCLR trained model. The choice of Llama2 for data generation was done considering its performances with respect to other models of the same size¹.

Next, we describe the generation and training steps in details.

¹<https://ai.meta.com/research/publications/llama-2-open-foundation-and-fine-tuned-chat-models/>

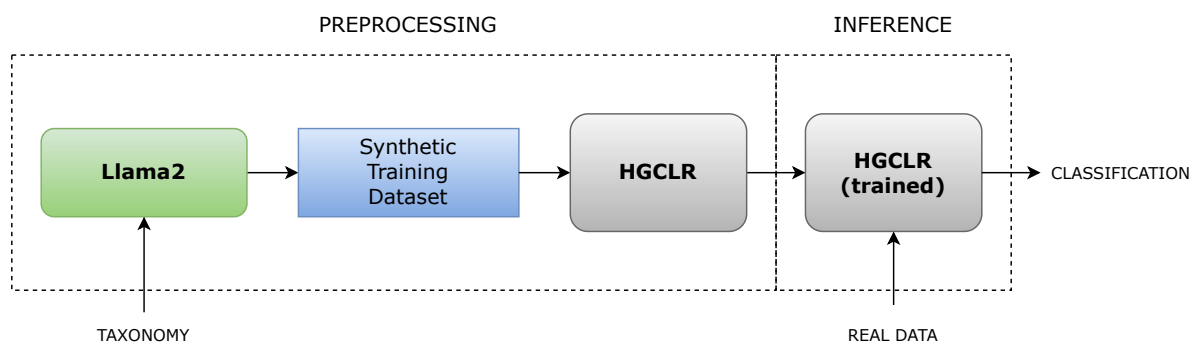


Figure 1: Functional schema of the HTC-GEN framework.

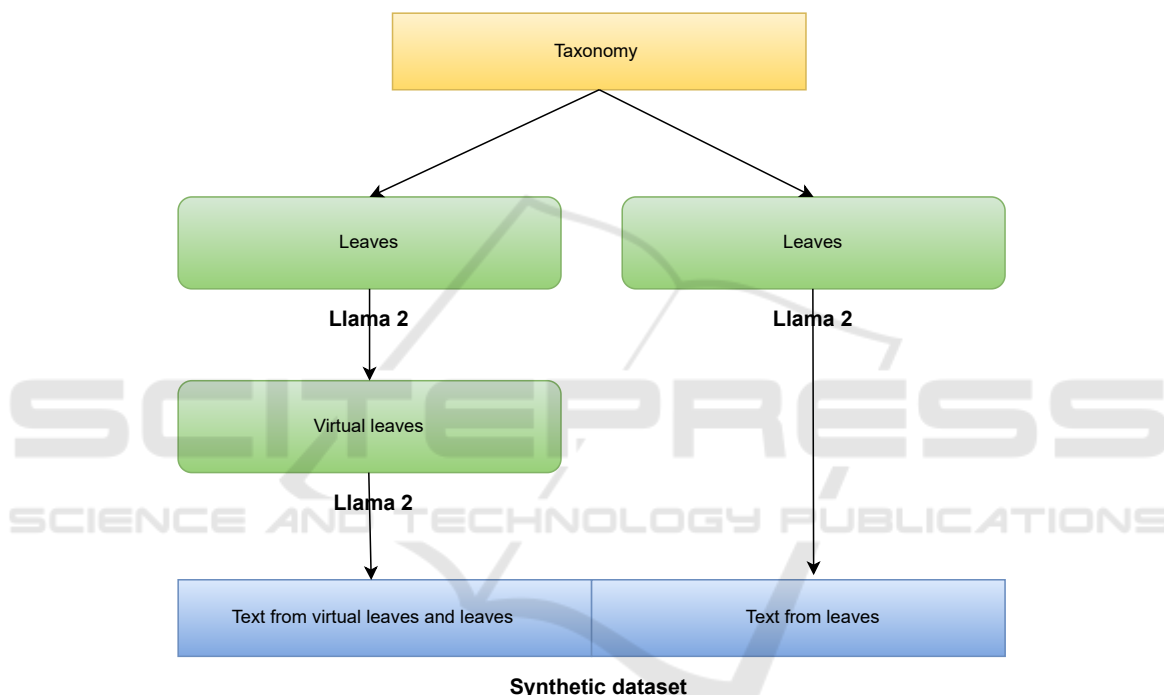


Figure 2: The pipeline of the Synthetic dataset creation starting from the target dataset's taxonomy.

3.1 Data Generation Module

The first module of the HTC-GEN framework (Fig. 2) focuses on the generation of synthetic data, by leveraging LLMs capabilities. We employ the LLM LLaMa-2², a collection of foundation language models ranging from 7B to 70B parameters, with checkpoints usually finetuned for chat applications.

Generally, LLaMa-2 responds more effectively to prompting strategies that involve providing general instructions in the “system” content for the desired outcome, and placing specific details in the “user” content in a meaningful sequence. The most significant details should be presented before the less im-

portant ones, considering their hierarchy. In light of the above, the general shape has to be as follows:

role: system
content: [GENERAL CONSTRAINTS]

role: user
content: [INSTRUCTIONS]

Algorithm 1 describes the steps to build a synthetic dataset by virtually extending the taxonomy with a further level of leaves as children of the starting taxonomy's leaves, here defined as *virtual leaves*. Such approach will increase variability of generated data to a greater extent from generating data starting

²<https://ai.meta.com/LLaMa/>

Algorithm 1: `generate_syn_dataset(taxonomy, vl_number, items_number, split_ratio, vl_name, item_name)`.

Input: (i) *taxonomy*: taxonomy of the target dataset (ii) *vl_number*: #virtual leaves to be generated for each *taxonomy* leaf (iii) *items_number*: #items to be generated for each *taxonomy* leaf (iiii) *size_ratio*: the size ratio between dataset from *leaves* and dataset from virtual leaves (iv) *vl_name*: virtual leaf name (v) *item_name*: target item name to be generated

Output: a new synthetic *dataset*

```

1: dataset ← [];
2: new_taxonomy ← taxonomy;
3: foreach leaf in new_taxonomy do
4:   virtual_leaves ← generate_leaves(vl_number,
   leaf, vl_name);
5:   leaf.append(virtual_leaves);
6: foreach leaf in new_taxonomy do
7:   foreach _ in range(items_number) do
8:     item ← generate_item(item_name, leaf);
9:     dataset.add(item);
10: size_two ← [vl_number · items_number · size_ratio]
11: foreach leaf in taxonomy do
12:   foreach _ in range(size_two) do
13:     item ← generate_item(item_name, leaf);
14:     dataset.add(item);
15: return dataset
    
```

from leaves. In Section 4.1 it is shown that such variability boosts also accuracy in the classification task. In line 1-2 of Algorithm 1 the variables *dataset* and *new_taxonomy* are initialized with empty set and the starting *taxonomy*, respectively, the latter given as an input parameter. In line 3-5, for each leaf of *new_taxonomy*, the function “generate_leaves” invokes an LLM by using the following prompt (**Prompt 1**) for virtual leaves generation, which are appended to *new_taxonomy*:

Prompt 1: generation of virtual leaves from a leaf

role: system

content: [VL_NAME] must be separated by commas

role: user

content: Generate [N] [VL_NAME] of [LEAF]

The triple (*vl_number*, *leaf*, *vl_name*) is given as parameters for ([N], [LEAF], [VL_NAME]) to **Prompt 1**, while system’s **content** is domain dependent and concerns the general morphology³ of what we aim to generate. Afterward in line 6-9, for each leaf of *new_taxonomy* the function “generate_item” invokes an LLM to generate *items_number* items which are added to *dataset*, by using the following prompt (**Prompt 2**) with the couple (*item_name*, *leaf*) as parameters for (ITEM_NAME, LEAF):

³For example: language, words separators, context, tone of text, etc.

Prompt 2: text generation from the leaf

role: system

content: the [ITEM_NAME] must be in English

role: user

content: Generate a [ITEM_NAME] from [LEAF]

where system’s **content** in **Prompt 2** is distinct from the one in **Prompt 1** since it refers to a different task. In line 10 the size (*size_two*) of a further dataset generated from *leaf* is calculated. The size considers a specific *size_ratio*, with $0 < size_ratio \leq 1$, which is given as an input parameter to create an additional synthetic dataset starting from *taxonomy*, considering *non-virtual* leaves, in order to increase data diversification by providing further semantic coverage outside the virtual leaves. In this way the final synthetic dataset will be a mixture of items from the two approaches. With *size_ratio*=0, the dataset would be generated totally from *virtual-leaves*; with a *size_ratio*=1 the two datasets would be equal in size (*vlaves_number* · *items_number*). In line 11-14 the function “generate_item” invokes again an LLM to generate a further dataset starting from *taxonomy* and sized above *size_two*, which is eventually merged to the dataset generated from virtual leaves. In line 15 the final synthetic dataset is returned.

In the case the dataset taxonomy is a Directed Acyclic Graph (DAG), further information about leaf parents must be included to distinguish leaves belonging to distinct branches. In this case, the user’s content in **Prompt 1** become: “Generate the [N] most frequent [VL_NAME] of [LEAF], [PARENT], [GRANDPARENT],...”, while the user’s **content** in **Prompt 2** become: “Generate a [ITEM_NAME] from [LEAF], [PARENT], [GRANDPARENT],...”. Such prompts modifications must be applied also to the input parameters of “generate_leaves” and “generate_items” in line 3, 8 and 13 of Algorithm 1.

A general heuristic for a proper Algorithm 1 sizing parameters choice is reported in Section 3.2.

To better explain the meaning of the variable parts in the prompts, we report in Table 1 possible values for these parts, in different context (Web of Science and other generic datasets). The column name “Leaf” is extracted from the lower level of the *taxonomy*.

Table 1: Possible values for the variable parts in the prompts in different contexts, i.e. Web of Science and other generic datasets.

Dataset	ITEM_NAME	VL_NAME
WoS	abstract	keyword
Movies	movie	sub-genre
Forum	post	sub-topic
News	news	sub-category

3.2 Synthetic Dataset Sizing

In order to maximize the impact of each synthetic generation approach for the task of text classification, we have to maximize the semantic *representativeness* of texts to be classified in the latent space of the model designed to such purpose. The more such representativeness, the more the chances of correct classifications. The diversity of inner data distribution plays an important role in such task, as well as data redundancy doesn't give any semantic coverage contributions. A popular approach (Cox et al., 2021) (although never tested along with hierarchical classification) for quantifying such *diversity* is with the *Remote Clique Score* (i.e., the average mean distance of a data instance to other instances) and the *Chamfer Distance Score* (i.e., the average minimum distance of a data instance to other instances). We propose to employ such metrics for real-data as gold standard, and size the two slices of synthetic datasets through the parameter *size_ratio* of Algorithm 1 (crf. Section 3.1), in order to achieve a Remote Clique Score (RCS) and Chamfer Distance Score (CDS) as close as possible to the gold standard. More formally, given D_{Real} , D_{Syn} collections of respectively *real*⁴ and *synthetic* text data, and RCS_{Real} , CDS_{Real} , RCS_{Syn} , CDS_{Syn} respectively RCS and CDS for such collections, we quantify $RCS-dist_{act}$ and $CDS-dist_{act}$ as follows:

$$RCS-dist_{act} = |RCS_{Real} - RCS_{Syn}|,$$

$$CDS-dist_{act} = |CDS_{Real} - CDS_{Syn}|,$$

reformulating also D_{Syn} as composition of two parts:

$$D_{Syn} = D_L \cup D_{VL},$$

where D_L is a collection synthetically generated from leaves, and D_{VL} is synthetically generated from *virtual* leaves (children of leaves).

In light of above the goal is to determine D_L and D_{VL} such that the dependent variables $RCS-dist_{act}$ and $CDS-dist_{act}$ are as close as possible to the following *ideal* values $RCS-dist_{best}$ and $CDS-dist_{best}$, where:

$$RCS-dist_{best} = \operatorname{argmin}_{D_{Real}, D_{Syn}} (|RCS_{Real} - RCS_{Syn}|),$$

$$CDS-dist_{best} = \operatorname{argmin}_{D_{Real}, D_{Syn}} (|CDS_{Real} - CDS_{Syn}|),$$

which means to find a good trade-off sizing between D_L and D_{VL} , determined empirically by computing $RCS-dist_{act}$ and $CDS-dist_{act}$ for each dataset after their generation, in order to choose the best D_{Syn} which minimize them. In any case, a semantic overlapping between D_L and D_{VL} is expected, which

⁴In scenarios where synthetic data are unavailable, we either explore alternative texts that closely resemble the distribution of the target dataset, or in presence of data to be classified, we select a sample among those.

causes information redundancy, but $CDS-dist_{act}$ in particular can constitute a valid indicator to evaluate such redundancy.

This approach is able to achieve a more meaningful semantic diversification of the generated data with respect to acting on the temperature parameter⁵, which tends to change only the sentences' morphology without really diversifying their meaning, hence it is able to significantly increase the performance of a classifier.

As for this work case-study, in Section 4.2 it is described how we chose the two slices of the synthetic dataset in order to improve the hierarchical classification.

3.3 HGCLR-Based Module

The second phase of our methodology involves using the previously generated synthetic dataset to fine-tune a supervised model specifically designed to solve the HTC task. To achieve this, we used HGCLR, which represents the current state-of-the-art model in the HTC field. The HGCLR model involves the use of a contrastive learning module, which includes the generation of positive samples, both label-driven and hierarchy-based. Furthermore, the method employs a Graphormer to encode the label hierarchy and produce label features. The HGCLR model incorporates a contrastive learning module, which encompasses the generation of positive samples using both label-driven and hierarchy-based techniques.

The initial steps of the model architecture comprise a BERT-based Text Encoder and a Graphormer-based Graph Encoder. These components are responsible for encoding and modeling the label hierarchy, respectively. The next step concerns generating positive samples that retain a fraction of tokens while preserving labels. In this technique, a token is retained for positive examples if its probability of being sampled exceeds a certain threshold, namely γ . Subsequently, the contrastive learning module works to make the encoded sentence-level representations of positive pairs as similar as possible, while ensuring that examples that are not from the same pair are distanced in the representation space. This is achieved through the application of a non-linear layer to the hidden states of positive pairs. For every batch, the NT-Xent loss (Chen et al., 2020) is computed, with the hyperparameter τ set accordingly. The total contrastive loss is the mean loss across all examples. Finally, the label hierarchy is flattened, and the hidden

⁵Temperature is the randomness of the outputs. A high temperature means that if you ran the same prompt 100 times, the outputs would look very different.

feature is given as input to a linear layer. A sigmoid function is employed to calculate the probability for prediction. The ultimate loss function combines the classification loss of the original data, the classification loss of the constructed positive samples, and the contrastive learning loss weighted by the hyperparameter λ .

4 RESULTS AND EVALUATION

We conduct a comprehensive experimental analysis to evaluate the effectiveness of our HTC-GEN system in classifying scientific abstracts in an HTC setting. In Section 4.1 we present the overall performance of HTC-GEN on the WoS dataset. In Section 4.2, we provide detailed results of the synthetic dataset generation process described in Section 3.1.

4.1 HTC-GEN Overall Results

HTC-GEN offers an automated framework that minimizes human effort by combining zero-shot and supervised methodological approaches. To adequately assess the efficacy of our method, we conduct extensive experimental analysis on the domain of HTC for scientific abstracts, including a comparative evaluation against state-of-the-art HTC models in both supervised and zero-shot settings.

Our analysis explores how these models react to synthetic and real data inputs and examines their performance in both zero-shot and supervised scenarios, offering valuable insights into their capabilities. We test our method on the WoS dataset (Kowsari et al., 2017), which comprises nearly 50,000 research abstracts, each labeled with technical keywords representing specific areas of research. The dataset taxonomy exhibits a two-level hierarchy, consisting of 7 labels at Level 1 and 134 labels at Level 2.

In a zero-shot scenario, we compare the HTC-GEN system 3 with the state-of-the-art model for the WoS dataset (i.e. Z-STC) (Bongiovanni et al., 2023). We train the HGCLR-based module of the model using the synthetic WoS dataset generated via LLaMa-2 (see Sect. 3.1). To generate our synthetic dataset we employed Algorithm 1 by calling `generate_syn_dataset` with the following parameters:

- *taxonomy* = WoS taxonomy
- *vl_number* = 20 (#virtual leaves as new children for each WoS taxonomy’s leaf)
- *items_number* = 10 (#items for each leaf in the synthetic dataset from virtual leaves)

- *size_ratio* = 0.5 (ratio between synthetic dataset from virtual leaves and synthetic dataset from leaves)
- *vl_name* = "keyword" (virtual leaf name)
- *item_name* = "abstract" (item name)

In this way the generated dataset will be made of 40200 items. For the training phase, we configured the parameters of the HGCLR-based module with Adam as optimizer with a learning rate of $3e - 5$ and a batch size of 12 for a total of 10 epochs. The threshold γ is set to 0.02, the contrastive loss weight λ set to 0.005 and the temperature τ of contrastive module set to 1.

The Z-STC method leverages the semantic information embedded in pre-trained Deep Language models to assign a prior relevance score to each label in the taxonomy using a zero-shot approach. Furthermore, it makes use of the hierarchical structure to support the preliminary result. As a baseline, we consider the high-performance version of the Z-STC method, which is based on the Semantic Text Embedding (STE) MPNet model⁶.

Table 2 shows the overall performance for both levels of the WoS taxonomy (i.e. Level 1 and Level 2) in terms of F1-score (macro). Notably, HTC-GEN outperforms the Z-STC model in both taxonomy levels, achieving F1-scores of 0.75 and 0.51, respectively.

In a supervised scenario, we conducted a comparative assessment between our model and the state-of-the-art for the second-level taxonomy of the WoS dataset (i.e., HGCLR) (Wang et al., 2022). To evaluate performance concerning synthetic data, we test HTC-GEN and HGCLR in both in- and out-of-distribution settings. In the In-Distribution scenario (ID), we test the models on data that mirrors the same distribution as their respective training sets. This involved fine-tuning the models with real data and subsequently evaluating their performance on real data (i.e. Train Real-Test Real), as well as fine-tuning the models on synthetic data and assessing their performance on synthetic data (i.e. Train Syn-Test Syn). Conversely, the Out-of-Distribution (OOD) scenario involved testing models on data with a distribution different from that of their training sets. This scenario involves fine-tuning the models with real data and subsequently evaluating them on synthetic data (i.e. Train Real-Test Syn), as well as fine-tuning the models on synthetic data and evaluating their performance on real data (i.e. Train Syn-Test Real). As

⁶<https://huggingface.co/sentence-transformers/all-mpnet-base-v2>

shown in Table 5, both models achieve higher performance in an ID setting with an F1-score of 0.97 and 0.84, respectively. In contrast, the system HTC-GEN shows a drop in performance when we test it in an OOD scenario with respect to the HGCLR model. This result is expected since synthetic data are generated with a different process than real data. We note that the drop in performance is more pronounced when the model trained on synthetic data is applied to real data (Macro F1 0.51). This is likely due to the fact that real data are harder to classify w.r.t synthetic data, where we observe less variability. Despite the drop in performances, the model trained on synthetic data still outperforms existing zero-shot methods and hence is valuable when training data are not available.

A final aspect of our experimental analysis focuses on the study of our model’s different components. In Table 4, we present insightful data regarding the influence of the HGCLR-based module 3.3 on the HTC-GEN system. To assess the impact of the HGCLR module on our system’s overall performance, we compare HTC-GEN with the Flat-GEN model, a RoBERTa-based multiclass classifier fine-tuned using the same dataset as HTC-GEN. Notably, Flat-GEN treats the HTC task as a flat classification task, encompassing all the 134 labels in the second level of the WoS taxonomy. We infer first-level labels from second-level predictions.

The results reveal a significant impact of model architecture on tool performance, with our method outperforming the Flat-GEN model by 7 and 4 percentage points for Levels 1 and 2 of the taxonomy, respectively.

Finally, we investigate the performance of HTC-GEN at different training sizes (number of items per virtual leaf). As shown in Table 3, the trend presents oscillations, ranging from a minimum of 0.746 to a maximum of 0.768 for the first level of the taxonomy, and from a minimum of 0.509 to a maximum of 0.516 for the second level. Overall, the results demonstrate a notable degree of stability. The performance trend is due to the inherent adaptability of the HGCLR-based module, which converges even with a limited amount of training data samples.

Table 2: Comparison between our method (i.e. HTC-GEN) and the state-of-the-art zero-shot Z-STC model for both Level 1 and Level 2 of the WoS taxonomy in terms of macro F1-score.

Model	WoS	
	Level 1	Level 2
Z-STC	0.74	0.46
HTC-GEN	0.75	0.51

Table 3: Performance of HTC-GEN model at different training set sizes (number of items per class).

Dataset size	WoS	
	Level 1	Level 2
10	0,768	0,516
20	0,752	0,513
50	0,746	0,509
100	0,748	0,516
All	0,749	0,513

Table 4: Overall results of HTC-GEN and Flat-GEN model on the WoS dataset in terms of macro F1-score.

Model	WoS	
	Level1	Level2
Flat-GEN	0.68	0.47
HTC-GEN	0.75	0.51

4.2 Synthetic Data Generation Performance

To test the first module of our methodology 3.1 we generate synthetic data using the LLM LLaMA-2 starting from the taxonomy of scientific paper abstracts provided by the WoS dataset.

In order to experimentally validate the impact of our synthetic generation approach in Section 3.1, we generated a number of synthetic datasets with different size and composition, and compute their Average Chamfer Distance Score, their Average Remote Clique Score (AVG CDS/RCS in Table 6) and their classification performances achieved by Flat-GEN (cf. Table 7), and hence without employing the downstream HGCLR-based module. The results show that the best F1-Score for the classification task on level 2 of WoS dataset is achieved in correspondence with the highest AVG RCS (the higher AVG RCS, the more information diversification), and the lowest information overlapping, denoted by values close to zero for AVG CDS (the lower AVG CDS, the more information redundancy).

Furthermore, the union of datasets generated from synthetic keywords and from classes (ZS-KWS 20 + ZS-100 in Table 6) boosts F1-Score to the same value (0.47) of the dataset achieved with *real* keywords from WoS dataset (ZS-KWS 10 GOLD in Table 6), which are supposed not to be available in *real* use cases.

The dataset (ZS-KWS 20 + ZS-100 in Table 6 and Table 7) employed for the HTC task includes a total of 40200 items, with 300⁷ items assigned to each label at the second level of the taxonomy.

⁷Which are 200 generated from virtual leaves, 100 generated from leaves.

Table 5: Performance Comparison of HTC-GEN and HGCLR models in In- and Out-of-Distribution scenarios for Level 2 of the taxonomy. In the In-Distribution (ID) setting, the "Train Real-Test Real" cell represents the performance of the HGCLR model, which is trained and tested on real data. Meanwhile, the "Train Syn-Test Syn" cell displays the outcomes of the HTC-GEN model, trained and tested on synthetic data. For the Out-of-Distribution (OOD) scenario, the "Train Real-Test Syn" cell shows the performance of the HGCLR model, trained on real data and tested on synthetic data. In contrast, the "Train Syn-Test Real" cell reports the outcomes of the HTC-GEN model, trained on synthetic data and tested on real data.

	Test Real	Test Syn
Train Real	0,81	0,84
Train Syn	0,51	0,97

Table 6: Comparison using Average Chamfer Distance Score (AVG CDS) and Average Remote Clique Score (AVG RCS) between real data distribution (WoS), Zero-Shot from WoS 10 *real* keywords/area (ZS-KWS 10 GOLD), Zero-Shot from 10/20 synthetic keywords/area (ZS-KWS 10/ZS-KWS 20), Zero-Shot from area (ZS-100) and the union of the last two (ZS-KWS 20 + ZS-100).

Dataset	AVG CDS	AVG RCS
WoS	0.000201	0.000989
ZS-KWS 10 GOLD	≈ 0	0.000706
ZS-KWS 10	0.000123	0.000654
ZS-KWS 20	0.000125	0.000656
ZS-100	≈ 0	0.000627
ZS-KWS 20 + ZS-100	≈ 0	0.000703

5 DISCUSSION

The results of our study offer an overview of the performance of the HTC-GEN model in the context of HTC in scientific abstracts. A key point of our methodology lies in its automation, which significantly reduces the dependence on human effort. HTC-GEN introduces an automated framework that minimizes human intervention in HTC, achieving this by joining zero-shot and supervised methodological approaches and leveraging the capabilities of LLMs in a cost-effective manner. Automated data generation proves especially valuable in scenarios where creating meticulously curated datasets requires a lot of time and labor. The performance achieved by the HTC-GEN model in a zero-shot scenario supports these considerations. Our model outperforms the state-of-the-art Z-STC model at both Level 1 and Level 2 of the WoS taxonomy. Furthermore, our model leverages the latest state-of-the-art supervised model for HTC classification, incorporating it into the architecture, which achieves stability and adaptability across different sizes of training datasets. This feature is of particular importance in real-world applications

Table 7: Comparison for non-hierarchical classification using *xml-roberta-large* between datasets in Table 6.

Dataset	Items/Class	F1-Score
WoS	274 (AVG)	0.67
ZS-KWS 10 GOLD	100	0.47
ZS-KWS 10	100	0.43
ZS-KWS 20	200	0.45
ZS-100	100	0.39
ZS-KWS 20 + ZS-100	300	0.47

where the availability of training data may be variable.

In conclusion, HTC-GEN presents a comprehensive, automated, and robust framework for HTC. Its benefits range from reducing human effort to achieving superior performance in different scenarios, supported by improved model architecture and adaptability. The broader implication of these advantages is that HTC-GEN represents a valuable resource in research and practical applications, particularly in fields characterized by dynamic data, resource-intensive data labeling, and a pressing need for automated text classification.

6 LIMITATIONS

The known limitation in this approach is that possibly the dataset taxonomy does not allow for an extension that results in appreciable benefits, particularly concerning the diversity of the newly generated data. However, it still represents a valid tool for non-hierarchical and hierarchical classification in scenarios of data scarcity or total absence of data.

7 CONCLUSION AND FUTURE WORKS

In this paper, a novel approach to address the task of hierarchical texts classification, in absence of training data was presented. Our approach tackles the issue that common methods for generating synthetic data involving Large Language Models lack the necessary variability of generated text. Thus we introduce a novel concept: expanding virtually the taxonomy by creating new children from leaves, thereby enabling the generation of synthetic data based on this newly *extended* taxonomy. The approach follows an insight aimed to grab both detailed and general semantic matching with the real data, by leveraging two metric from the state-of-the-art useful to quantify *diversity* and *redundancy* on text distributions.

To validate the methodology, the case-study of the

WoS dataset was considered, by invoking Llama2-7B-chat through a specifically designed Algorithm based on zero-shot prompting.

In regard of HTC task, we employed the Hierarchy Guided Contrastive Learning model, which represents the current state-of-the-art in the field of supervised HTC. The experiments conducted by training the above model on a synthetic dataset and tested with real data extracted from WoS, showed that our approach HTC-GEN outperforms the current ZERO-shot state-of-the-art approach.

For future works we plan to test our approach on more datasets, with bigger models than Llama2-7B-chat and also by integrating other techniques (Chung et al., 2023) from the state-of-the-art aiming to refine further the quality of the synthetic dataset.

DISCLAIMER

The content of this article reflects only the author's view. The European Commission and MUR are not responsible for any use that may be made of the information it contains.

ACKNOWLEDGEMENTS

This work was supported by FOSSR (Fostering Open Science in Social Science Research), funded by the European Union - NextGenerationEU under NRRP Grant agreement n. MUR IR0000008.

REFERENCES

- Bhambhoriya, R., Chen, L., and Zhu, X. (2023). A simple and effective framework for strict zero-shot hierarchical classification.
- Bongiovanni, L., Bruno, L., Dominici, F., and Rizzo, G. (2023). Zero-shot taxonomy mapping for document classification. In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, pages 911–918.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.
- Chung, J., Kamar, E., and Amershi, S. (2023). Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.
- Cox, S. R., Wang, Y., Abdul, A., von der Weth, C., and Y. Lim, B. (2021). Directed diversity: Leveraging language embedding distances for collective creativity in crowd ideation. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21. ACM.
- Haj-Yahia, Z., Sieg, A., and Deleris, L. A. (2019). Towards unsupervised text classification leveraging experts and word embeddings. In *Proceedings of the 57th annual meeting of the Association for Computational Linguistics*, pages 371–379.
- Jeronymo, V., Bonifacio, L., Abonizio, H., Fadaee, M., Lotufo, R., Zavrel, J., and Nogueira, R. (2023). Inpars-v2: Large language models as efficient dataset generators for information retrieval. *arXiv preprint arXiv:2301.01820*.
- Ko, Y. and Seo, J. (2000). Automatic text categorization by unsupervised learning. In *COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics*.
- Kowsari, K., Brown, D. E., Heidarysafa, M., Meimandi, K. J., Gerber, M. S., and Barnes, L. E. (2017). Hdltext: Hierarchical deep learning for text classification. In *2017 16th IEEE international conference on machine learning and applications (ICMLA)*, pages 364–371. IEEE.
- Liu, R., Liang, W., Luo, W., Song, Y., Zhang, H., Xu, R., Li, Y., and Liu, M. (2023). Recent advances in hierarchical multi-label text classification: A survey. *arXiv preprint arXiv:2307.16265*.
- Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Stammach, D. and Ash, E. (2021). Docscan: Unsupervised text classification via learning from neighbors. *arXiv preprint arXiv:2105.04024*.
- Wang, Z., Wang, P., Huang, L., Sun, X., and Wang, H. (2022). Incorporating hierarchy into text encoder: a contrastive learning approach for hierarchical text classification.
- Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. (2021). Fine-tuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.
- Zangari, A., Marcuzzo, M., Schiavinato, M., Rizzo, M., Gasparetto, A., Albarelli, A., et al. (2023). Hierarchical text classification: a review of current research. *EXPERT SYSTEMS WITH APPLICATIONS*, 224.
- Zhang, Y., Yang, R., Xu, X., Xiao, J., Shen, J., and Han, J. (2024). Teleclass: Taxonomy enrichment and llm-enhanced hierarchical text classification with minimal supervision. *arXiv preprint arXiv:2403.00165*.