Jump To Hyperspace: Comparing Euclidean and Hyperbolic Loss Functions for Hierarchical Multi-Label Text Classification

Jens Van Nooten and Walter Daelemans
CLiPS (University of Antwerp)
Prinsstraat 13, 2000 Antwerp (Belgium)
firstname.lastname@uantwerpen.be

Abstract

Hierarchical Multi-Label Text Classification (HMLTC) is a challenging machine learning task where multiple labels from a hierarchically organized label set are assigned to a single text. In this study, we examine the effectiveness of Euclidean and hyperbolic loss functions to improve the performance of BERT models on HMLTC, which very few previous studies have adopted. We critically evaluate label-aware losses as well as contrastive losses in the Euclidean and hyperbolic space, demonstrating that hyperbolic loss functions perform comparably with non-hyperbolic loss functions on four commonly used HMLTC datasets in most scenarios. While hyperbolic label-aware losses perform the best on low-level labels, the overall consistency and micro-averaged performance is compromised. Additionally, we find that our contrastive losses are less effective for HMLTC when deployed in the hyperbolic space than non-hyperbolic counterparts. Our research highlights that with the right metrics and training objectives, hyperbolic space does not provide any additional benefits compared to Euclidean space for HMLTC, thereby prompting a reevaluation of how different geometric spaces are used in other AI applications[1].

1 Introduction

Hierarchical Text Classification (HTC) is a text classification problem where the labels are organized as a hierarchical structure, often represented as a Directed Acycilic Graph (DAG) or tree. This type of classification problem has been widely researched for both image classification and text classification in single-label and multi-label settings. Most of the work surrounding HTC is concerned with optimally leveraging and representing the hierarchical structure of the labels for learning hierarchical text representations. Some research on hierarchical classification has indicated that the hyperbolic space is more suitable than the Euclidean to model hierarchical representations between features of texts or images, due to the space's constant negative curvature (Nickel and Kiela, 2017; Ganea et al., 2018; Gulcehre et al., 2018). Because of this curvature, the space allows nodes in a hierarchy to be spaced out more effectively. Even though a few studies have incorporated it for HTC (Chatterjee et al., 2021; Chen et al., 2023, 2020a), no other work has directly compared label-aware losses or contrastive losses in the Euclidean space and hyperbolic space with LLMs for Hierarchical Multi-Label Text Classification (HMLTC), to the best of our knowledge. The contributions of our work are the following:

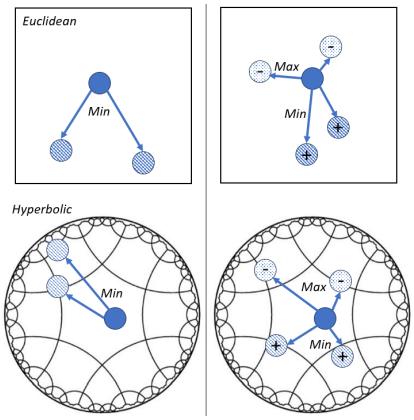

Figure 1: Our approach uses label-aware losses to minimize distance from true labels (left), and contrastive losses to minimize distance in positive pairs and maximize it in negative pairs (right).

- We thoroughly compare the impact of deploying several loss functions in the Euclidean and hyperbolic space on four commonly used HMLTC datasets.
- We introduce previously established label-aware losses and novel (hyperbolic) contrastive losses to HMLTC.

[1]The code is available on https://github.com/clips/jump_to_hyperspace.

Proceedings of the 31st International Conference on Computational Linguistics, pages 4260–4273
January 19–24, 2025.

We find that Euclidean and hyperbolic label-aware losses perform almost on-par with each other and that contrastive learning in the hyperbolic space is ineffective. While our study is limited to exploring NLP applications, it offers insights into the geometric underpinnings of learning algorithms which could influence broader AI applications such as knowledge graph embedding and multi-task learning for hierarchical representations.

2 Related Research

2.1 Hierarchical Multi-Label Text Classification

The main research question in HMLTC is how the hierarchical nature of the label sets can be optimally leveraged during training, or, how to optimally learn hierarchical text and label representations. Many different approaches have been taken to tackle HMLTC, though they can be generally split up in two main categories: Global (one flat classifier) and local (multiple classifiers) classification. The most recent approaches range from using label correlations (Xu et al., 2021; Zhang et al., 2021) and label-aware representations (Chen et al., 2020a; Zhou et al., 2020; Deng et al., 2021) to attention-based methods, such as label-based attention (Zhang et al., 2022a) or hierarchical attention (Lu et al., 2022). Additionally, Wang et al. (2023) opt for data augmentation with LLMs to improve model performance for HMLTC. Some studies have also investigated a multi-task approach, where models simultaneously learn to classify and minimize the distance between text and label representations within a shared embedding space (Chen et al., 2020a, 2021b), an approach that we aim to explore further with our work.

2.2 Training Neural Networks in the Hyperbolic Space

Hyperbolic Space While most neural networks are trained and fine-tuned in the Euclidean space, some research argues that the hyperbolic space is more preferable for learning hierarchical representations (Nickel and Kiela, 2017; Ganea et al., 2018). The hyperbolic space is a space defined by its constant negative curvature, as opposed to the flatness of the Euclidean space. The Euclidean space adheres to Euclidean axioms, such as the parallel postulate, meaning that parallel lines never converge. This contrasts with the hyperbolic space, where lines have a constant hyperbolic curvature, continually diverging as they extend. Consequently, the different nature of the hyperbolic space affects the way mathematical operations are performed, such as vector additions and measuring the distance between points (Ganea et al., 2018).

Poincaré Ball Model The Poincaré ball model (bottom row of Figure 1) is a widely-used model that represents the hyperbolic space within a sphere-like space (Nickel and Kiela, 2017; Ganea et al., 2018). The main intuition behind using the hyperbolic space –represented as this ball model– to effectively model hierarchical relationships is that the distance between two points on this space increases exponentially as points move away from the center. Hierarchies can be represented as trees where the number of nodes increases exponentially the more hierarchy levels the tree has, which naturally lends itself well to the constant negative curvature of the hyperbolic space. This allows the hierarchy nodes to be more efficiently "spaced out" compared to the flat surface of the Euclidean space, given the same spatial dimensions. Therefore, this way of embedding hierarchical nodes is postulated to be especially useful for representing low-level leaf nodes in a hierarchy (Nickel and Kiela, 2017; Ganea et al., 2018; Chen et al., 2023).

Hyperbolic Learning for Computer Vision and NLP Some works in the field of computer vision focus on leveraging the hyperbolic space for modelling hierarchical relationships between features (Dhall et al., 2020; Yue et al., 2023; Xiong et al., 2022), though very few works have explored the hyperbolic space for NLP tasks (Gulcehre et al., 2018; Chen et al., 2021a), especially for hierarchical text classification (Chen et al., 2020a; Chatterjee et al., 2021; Chen et al., 2023). Training Deep Neural Networks (DNNs) in general for text-based applications in the hyperbolic space was explored further by Nickel and Kiela (2017), who trained entailment representations of graph nodes using Poincaré embeddings. Ganea et al. (2018) improved upon these results by training hierarchical node representations as hyperbolic entailment cones with a Graph Neural Network (GNN)[2], where the branches of a hierarchy are embedded as a cone shape in the hyperbolic or Euclidean space, thus allowing for a more well-defined discrimination between nodes and entire branches in an embedding space. The

[2]Essentially, the model is trained on a binary link prediction task ("Is node n a child node of m?").

authors found that representing nodes in the hyperbolic space –and as entailment cones– yields improved hierarchical representations.

Chen et al. (2020a) adopt the hyperbolic space for training label-aware text representations with GLoVE embeddings and a GRU encoder. They found that their Hyperbolic Interaction Model (HyperIM) outperformed the Euclidean counterpart on several HMLTC datasets, though no experiments were conducted with LLMs. To better model hierarchical representations of emotion words, Chen et al. (2023) took inspiration from HyperIM and entailment cones by minimizing the distance between hyperbolic BERT embeddings and the corresponding true labels (represented as hyperbolic entailment cones of an emotion graph), which lead to performance increases on several multi-class emotion classification datasets. However, no comparison was made with the same loss function deployed in the Euclidean space. Furthermore, Chatterjee et al. (2021) combine hyperbolic label representations with label correlations during training.

Conversely, Fivez et al. (2021) found that their approach of leveraging the cosine distance in the Euclidean space proved to be more effective than more complex hyperbolic methods for modelling biomedical hypernym relationships. The authors do so by explicitly grounding the model by minimizing the cosine distance between a concept representation and its prototypical representation. Given these findings, and due to the limited comparisons between the Euclidean and hyperbolic space for such loss functions in HMLTC with LLMs, we deploy our loss functions in both spaces.

2.3 Contrastive Learning for Multi-Label Classification

Contrastive Learning Contrastive Learning (CL) is a training method where models are encouraged to separate dissimilar representations and bring similar representations closer in an embedding space, leading to a better performance on downstream tasks (Chen et al., 2020b). CL has been well researched and has shown promising results for several single-label and multi-label NLP tasks (Zhang et al., 2022b; Yu et al., 2023). However, due to the complexity of the label space and, consequently, the semantic embedding space of multi-label instances, the notion of positive and negative examples for an anchor (i.e. a training instance) also becomes more complex. A positive example could be an instance with the exact same label set, though this is too restrictive on datasets with a large number of labels. Generally, given a batch of training samples, positive samples can be retrieved within a batch (Lin et al., 2023), generated based on an anchor (Lu et al., 2022) or by using class prototypes (Audibert et al., 2024). Lin et al. (2023) explored several sample selection methods for multi-label emotion classification on the SemEval datasets, including calculating the overlap between binary label vectors of the anchor and other in-batch examples using the Jaccard Index metric. This way, hard positives (all labels must overlap) or soft positives (only a percentage of labels must overlap) can be retrieved. In combination with the established Supervised Contrastive Loss (SCL) (Khosla et al., 2020), considerable improvements are achieved for multi-label emotion classification.

Contrastive Learning for Hierarchical Classification Several papers study contrastive learning for HMLTC by leveraging a multi-headed attention mechanism (Yu et al., 2023) or local contrastive learning (Chen et al., 2024). Additionally, multi-label CL in the hyperbolic space has been explored for image classification in Yue et al. (2023). In this paper, the authors found that minimizing and maximizing the distance between positive and negative pairs respectively in the hyperbolic space excels at supervised pretraining and classification, compared to SimCLR (Chen et al., 2020b), a Euclidean loss function. In the present work, we aim to fill the gap in the literature by further exploring the hyperbolic space for HMLTC with LLMs and making direct comparisons between the two geometric spaces for two sets of loss functions.

3 Methodology

We fine-tune models with two sets of loss functions that are adaptable to the Euclidean and hyperbolic space. The first set aims to minimize the distance between a text embedding and its true labels represented as hierarchical embeddings through entailment cones, based on earlier work by Chen et al. (2020a) and Chen et al. (2023). This set of loss functions will be referred to as Label-Aware losses (LA). The second set of losses are Multi-Label Contrastive Losses (MLCL), which aim to minimize and maximize the distance between similar and dissimilar pairs of texts respectively. The details of the datasets, evaluation metrics and loss functions are described in the following sections.

	BGC	RCV1-V2	AAPD	WOS
Text type	Book descs.	News articles	Abstracts	Abstracts
N train	58,715	20,826	53,840	30,070
N val	14,785	2,323	1,000	7,518
N test	18,394	781,265	1,000	9,397
N lbls. (lvls.)	146 (7, 36, 77, 7)	103 (4, 22, 33, 43, 1)	61 (9, 52)	150 (7, 143)
Avg lbls. / text	3.01	3.24	2.41	2
Max. lbl. count | Min. lbl. count	21,872 | 3	9,290 | 2	17,152 | 350	750 | 1
Corr. lbl. cnt. & lbl. lvl. |	-.48	-.42	-.37	-.87

Table 1: Statistics for each dataset. The last row depicts the Pearson correlation values between label frequency and label level.

3.1 Datasets

We compare the performance of the hyperbolic- and non-hyperbolic loss (i.e. Euclidean) function on four of the most commonly used datasets for HMLTC, namely the Blurb Genre Collection (BGC) (Aly et al., 2019), RCV1-v2 (Lewis et al., 2004), the Arxiv Academic Papers Dataset (AAPD) (Yang et al., 2018) and the WOS-46985 dataset (Kowsari et al., 2017). Details about each dataset can be found in Table 1. It should be noted that we augmented the AAPD dataset to include the nine highest level nodes, similar to Xu et al. (2021)[3].

3.2 Evaluation Metrics

We evaluate our models with macro- and micro-averaged F_1 in addition to the Exact Match Ratio (EMR), which deems a prediction set as correct when all labels are predicted correctly. Finally, we calculate the hierarchical precision, recall and F_1 (Kiritchenko et al., 2005), which are micro-averaged scores on prediction sets to which all ancestor nodes are added. This way, models that predict leaf nodes from other branches are punished more severely.

3.3 Label-Aware Losses

Label Nodes as Entailment Cones The first set of losses that we utilize is a set of label-aware losses (LA), based on previous work (Chen et al., 2023, 2020a). A visualization of this approach can be found in Figure 1. We first train label embeddings with a GNN as entailment cones (Chen et al., 2023; Ganea et al., 2018) both in the Euclidean space and the hyperbolic space for a binary link prediction task. For the hyperbolic space, we adopt the Poincaré ball model to represent the labels, thereby following the original implementation. The label dimension is set to 100[4].

Label-Aware Fine-tuning Once these label embeddings are obtained, BERT is fine-tuned to minimize the Binary Cross-Entropy Loss and to minimize the distance between the text representation and their respective hierarchical label embeddings, following Chen et al. (2023). With this approach, we aim to combine the contextual semantic knowledge of BERT embeddings with the explicit hierarchical information encoded in the label embeddings. For Euclidean embeddings, we use cosine distance or Euclidean distance as a distance measure, while we use the Poincaré distance –the most commonly used hyperbolic stance measure– for hyperbolic label representations. The Poincaré distance between hyperbolic vectors *u* and *v* is defined as follows, where $\|u\|$ and $\|v\|$ are the Euclidean norms of u and v, and *arcosh* is the inverse hyperbolic cosine function:

$$d(u,v) = \operatorname{arcosh}\left(1 + 2\frac{\|u-v\|^2}{(1-\|u\|^2)(1-\|v\|^2)}\right) \quad (1)$$

Before calculating the hyperbolic distance, we must assure that the text embedding is projected to the hyperbolic space by training an additional set of weights that learn to map vectors from the Euclidean space to the hyperbolic space by means of an exponential mapping (Chen et al., 2023)[5]. Let *p* denote a point on a n-dimensional Poincaré ball model of the hyperbolic space, $\oplus$ the Möbius addition and *v* a tangent vector at *p*:

$$\exp_p(v) = p \oplus \left(\tanh\left(\frac{\lambda_p\|v\|}{2}\right)\frac{v}{\|v\|}\right) \quad (2)$$

BERT embeddings are projected from their original hidden size to the same hidden size as the label embeddings. The label-aware loss L_{LA} is calculated as follows: Given is a batch with *N* items, with *M* labels per item, an embedding y_{ij} for label *j* assigned to the i^{th} item in the batch of (projected) text embeddings *x*. The Poincaré (or Euclidean) distance *d* is calculated between x_i and y_{ij}. The resulting distance is scaled with hyperparameter α[6]

[3]Even though this dataset is hierarchical in nature, the original version does not include the first-level nodes in the data

[4]Other dimensions (300 and 768) showed negligible performance differences.

[5]This mapping adds 77k trainabale parameters.

[6]By introducing this hyperparameter, the model is not "distracted" from learning the classification task itself by assigning a lower value to the secondary loss. Conversely, this hyperparameter can also increase the effect of the secondary task during training.

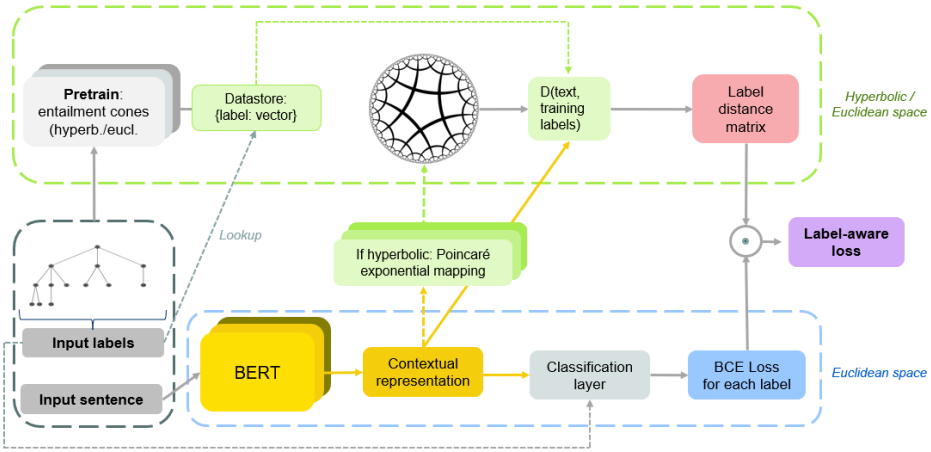

Figure 2: Visualisation of the Label-Aware (LA) training process.

and then multiplied with the Binary Cross-Entropy (L_{BCE}) loss for the corresponding labels Y_{ij}:

$$L_{\text{LA}} = \frac{1}{N}\sum_{i=1}^{N}\left(\alpha \times (\frac{1}{M_i}\sum_{j=1}^{M_i} d(x_i, y_{ij})) \times L_{\text{BCE}}(Y_{ij})\right) \tag{3}$$

3.4 Contrastive Losses

Pos. and Neg. Sample Selection Contrastive learning for multi-label text classification is especially challenging since the label space and embedding space are complex. Figure 3 visualizes our proposed contrastive loss. We propose to select positive and negative samples based on the Jaccard Index (JI) of binary –or multi-hot– representations of label vectors[7], following Lin et al. (2023). A sample is considered a positive when the JI is equal to or greater than a predefined threshold, namely 0.5. We hypothesize that soft positives (JI < 1) are a valid approach in the context of HMLTC, since texts from the same branch but with different leaf nodes ought to be clustered more closely together than texts from a different branch.

Multi-Label CL State-of-the-art contrastive losses usually leverage a variant of Supervised Contrastive Loss (Khosla et al., 2020; Yu et al., 2023), though we opt for a variant that does not necessarily require positive samples for an anchor in a batch. Since positive sample generation remains an open issue and there is no guarantee that a positive sample for each item is present in mini-batches due to the large number of labels in HMLTC datasets[8], we construct pairs by taking all possible combinations between items in a batch to extract as much information out of a single batch as possible. We adopt a margin-based contrastive loss (CL) function inspired by Sentence Transformers (Reimers and Gurevych, 2019) that minimizes the distance between similar pairs and maximizes the distance between negative pairs using a predefined distance measure, which in our case are either cosine distance, Euclidean distance or Poincaré distance. Given a batch with individual pairs *(u, v)*, binary label γ assigned to a pair (1 for positive, 0 for negative), a distance function d[9] and a margin θ, this loss function can be expressed as follows for a single text pair in a minibatch:

$$\begin{aligned} L_{\text{MLCL}} = \frac{1}{N}\sum_{i=1}^{N}\Big[&\gamma \cdot \text{d}(u,v)^2 \\ &+ \\ &(1-\gamma)\cdot \text{ReLu}(\theta - \text{d}(u,v))^2\Big] \end{aligned} \tag{4}$$

The first part of the loss function is activated when a pair is positive ($\gamma = 1$) , while the second part is activated when a pair is negative ($\gamma = 0$). The loss yields 0 when a negative pair's distance is equal to or less than the predefined margin. The final loss for a batch can be expressed as the sum of L_{MLCL}, which is weighted by hyperparameter α[10], and the Binary Cross-Entropy (L_{BCE}) loss for a batch:

[7] Binary vectors of length *N*, representing each unique label in the dataset with 1 for presence and 0 for absence.

[8] In Yu et al. (2023), the authors use a batch size of 80, which allows them to fully leverage SupConLoss, since uncommon labels are more likely to co-occur in a batch.

[9] For training stability, we divide the Poincaré and Euclidean distance by 10.

[10] Similar to L_{LA},α puts less or more emphasis on the secondary loss during training.

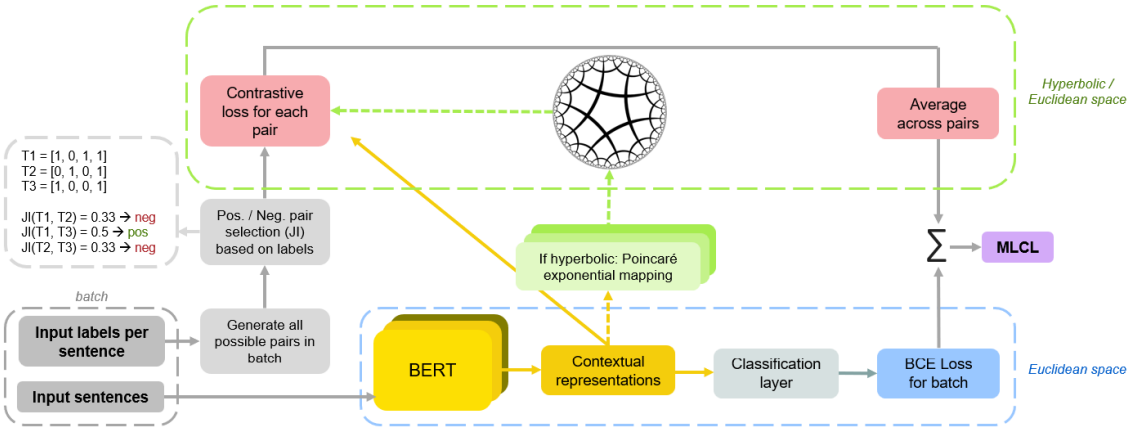

Figure 3: Visualisation of the Multi-Label Contrastive Loss (MLCL) training process.

$$L_{\text{Final}} = (\alpha \times L_{\text{MLCL}}) + L_{\text{BCE}} \quad (5)$$

Similar to the Label-Aware Loss, we project the text embeddings u and v to the hyperbolic space using an exponential mapping before calculating the Poincaré distance.

3.5 Classification Models

As classification models we employ BERT (Devlin et al., 2019) and DistilBERT (Sanh et al., 2020)[11], but the loss functions that we use can be employed in conjunction with any other type of feature encoder. The former model is used because of its success in HMLTC (Jiang et al., 2022; Wang et al., 2022), while the latter model's performance has been underexplored in HMLTC, even though the smaller model size is more attractive for industrial applications.

We grid search the optimal learning rate for each method and α hyperparameters for the label-aware losses and contrastive losses. All hyperparameters are summarized in Table 4 and 5 in Appendix A. For all experiments, we used the largest batch size that fit within the working memory, which was 8 with 2 gradient accumulation steps. We use the Adam optimizer and a learning rate scheduler with linear decay. All models are fine-tuned for 15 epochs, except for on the WOS dataset, on which we fine-tune the models for 10 epochs[12]. We repeat each experiment five times with a different random seed and report the average scores.

For training label embeddings as entailment cones (LA loss), we follow the original implementation by Ganea et al. (2018) and Chen et al. (2023) by setting the learning rate to $1e^{-3}$ and the label dimension size to 100. We train the models for 1,000 epochs. All experiments were conducted on NVIDIA GeForce RTX 2080 Ti GPUs with 11,264GB of RAM.

4 Results and Discussion

Euclidean vs. Hyperbolic losses The results, as summarized in Table 2[13], show consistent trends across different base encoders and datasets, in that both spaces generally produce comparable results. This is especially the case for the label-aware losses, where we only observe a .03 and .09 difference in macro and micro-averaged F_1 respectively on the BGC dataset, for example. Concerning the contrastive losses, we observe that hyperbolic MLCL underperforms compared to the cosine-based MLCL, but performs similarly to MLCL with Euclidean distance on all datasets except AAPD. Differences of up to 1 macro-averaged and 0.8 micro-averaged F_1 point are observed on the datasets.

When investigating the performance of the models per hierarchical level (Table 3), we again observe that geometric space only makes a slight difference. For the LA-models, performance increases of up to 7 macro F_1 points are observed at the lowest levels compared to the baselines. It is also important to note that the non-hyperbolic LA models yield virtually the same results on almost all levels as the hyperbolic LA models. For exam-

[11] *bert-based-cased* and *distilbert-base-cased* on `https://huggingface.co/`.

[12] Preliminary experiments showed that 10 epochs yielded equal, if not better results on the validation set than fine-tuning for 15 epochs.

[13] The results for DistilBERT can be found in Table 6, Appendix B.

	BGC						RCV1						AAPD						WOS					
Setup	maF_1	miF_1	EMR	hPr	hR	hF_1	maF_1	miF_1	EMR	hPr	hR	hF_1	maF_1	miF_1	EMR	hPr	hR	hF_1	maF_1	miF_1	EMR	hPr	hR	hF_1
Chat-GPT (Resuts from Yu et al. (2023))	35.63	57.17	/	/	/	/	32.30	51.35	/	/	/	/	45.82	27.98	/	/	/	/	/	/	/	/	/	/
BERT	62.57 (.32)	79.7 (.11)	47.08 (.31)	81.12 (.19)	78.2 (.17)	79.63 (.11)	54.16 (1.48)	79.94 (.45)	55.56 (.74)	82.78 (.33)	77.35 (.53)	79.97 (.41)	58.02 (.54)	81.49 (.16)	40.42 (.61)	84.14 (.35)	79.02 (.15)	81.5 (.16)	75.94 (.43)	86.39 (.12)	77.96 (.28)	87.95 (.12)	84.87 (.14)	86.39 (.11)
BERT + LA (hyperb.)	**65.15** (.19)	79.73 (.06)	46.24 (.12)	77.24 (.16)	82.14 (.16)	79.71 (.06)	**60.35** (.74)	80.11 (.38)	55.12 (.73)	77.79 (.39)	82.29 (.13)	79.98 (.21)	**60.07** (.68)	80.58 (.42)	37.46 (1.46)	78.48 (.76)	82.74 (.36)	80.55 (.44)	**77.92** (.27)	86.46 (.26)	77.64 (.39)	85.07 (.3)	**87.9** (.17)	86.46 (.23)
BERT + LA (eucl.)	65.12 (.28)	79.62 (.12)	46.08 (.27)	77.41 (.1)	**82.29** (.28)	79.78 (.08)	60.03 (.18)	79.87 (.19)	54.76 (.36)	77.49 (.56)	**82.33** (.23)	79.83 (.19)	59.25 (.56)	80.04 (.68)	36.5 (1.73)	77.33 (1.52)	**82.85** (.53)	79.98 (.68)	77.18 (.41)	86.27 (.48)	77.18 (.22)	84.73 (.07)	87.88 (.06)	86.27 (.04)
BERT + LA (cos.)	64.44 (.39)	80.01 (.23)	47.65 (.24)	80.08 (.27)	80.12 (.21)	80.1 (.21)	58.82 (.57)	80.57 (.17)	56.51 (.17)	81.3 (.13)	79.85 (.17)	80.94 (.08)	58.77 (.94)	81.02 (.69)	38.22 (1.78)	80.53 (1.4)	81.41 (.41)	80.88 (.71)	76.83 (.42)	86.31 (.26)	78.22 (.41)	86.52 (.18)	86.14 (.14)	86.33 (.16)
BERT + MLCL (hyperb.)	62.26 (.59)	79.96 (.13)	47.38 (.27)	81.28 (.19)	78.68 (.16)	79.96 (.11)	54.64 (.4)	80.54 (.13)	56.19 (.21)	83.64 (.14)	77.58 (.17)	80.5 (.11)	58.51 (.25)	**81.84** (.16)	**41.6** (.48)	**84.68** (.28)	79.27 (.31)	**81.89** (.13)	76.79 (.29)	86.67 (.11)	78.37 (.13)	88.07 (.07)	85.32 (.15)	86.67 (.1)
BERT + MLCL (Eucl.)	58.43 (.58)	79.45 (.13)	46.54 (.31)	81.56 (.13)	77.46 (.16)	79.46 (.13)	52.25 (.19)	80.0 (.07)	55.04 (.16)	84.21 (.08)	76.23 (.09)	80.02 (.07)	58.77 (.94)	81.02 (.69)	38.22 (1.78)	83.19 (.32)	78.54 (.27)	80.8 (.29)	76.41 (.33)	87.21 (.13)	76.41 (.33)	88.91 (.13)	85.57 (.13)	87.21 (.13)
BERT + MLCL (cos)	62.53 (.39)	**80.42** (.07)	**47.95** (.34)	**82.25** (.26)	78.68 (.09)	**80.42** (.17)	55.61 (.34)	**80.89** (.07)	**57.04** (.09)	**84.46** (.3)	77.69 (.19)	**80.94** (.08)	58.75 (1.21)	81.75 (.43)	41.2 (.6)	84.53 (.27)	79.12 (.67)	81.74 (.47)	77.82 (.29)	**87.35** (.1)	**79.13** (.21)	**88.96** (.14)	85.79 (.05)	**87.35** (.09)

Table 2: Results from all methods with BERT across each dataset, showing macro F_1, micro F_1, Exact Match Ratio, hierarchical precision, recall and F_1 scores with standard deviations across random seeds. 'Cos' refers to the cosine distance in the Euclidean space. The best results across models are marked in bold.

	BGC				RCV1					AAPD		WOS	
Setup	L1	L2	L3	L4	L1	L2	L3	L4	L5	L1	L2	L1	L2
BERT	83.3 (.32)	59.97 (.2)	59.48 (.78)	49.05 (.68)	69.74 (.17)	28.51 (.86)	60.39 (.6)	50.62 (.92)	87.07 (1.88)	69.09 (.85)	56.1 (.58)	91.19 (.15)	75.19 (.39)
BERT + LA (hyperb.)	83.75 (.12)	61.11 (.14)	63.14 (.28)	**55.4** (1.18)	**70.28** (.12)	**34.22** (.16)	**64.24** (.4)	**57.05** (.34)	**89.82** (1.51)	**71.48** (1.35)	**58.1** (.63)	91.63 (.24)	**77.25** (.25)
BERT + LA (eucl.)	83.87 (.32)	**61.13** (.14)	**63.33** (.26)	54.48 (.87)	70.06 (.12)	33.82 (.34)	64.19 (.16)	57.0 (.36)	89.73 (.73)	70.59 (1.13)	57.29 (.54)	91.47 (.14)	76.48 (.39)
BERT + LA (cos.)	83.73 (.21)	61.01 (.19)	62.6 (.38)	51.49 (2.22)	70.19 (.1)	33.14 (.29)	63.77 (.38)	54.97 (.45)	89.61 (.66)	67.98 (.53)	57.17 (1.07)	91.34 (.08)	76.05 (.37)
BERT + MLCL (hyperb.)	83.72 (.18)	60.45 (.18)	59.44 (.72)	48.93 (1.69)	70.01 (.13)	29.07 (.81)	60.7 (.31)	50.69 (.32)	88.56 (.12)	68.14 (1.06)	56.85 (.22)	91.37 (.16)	76.08 (.27)
BERT + MLCL (Eucl.)	83.07 (.21)	59.35 (.19)	53.98 (.52)	44.13 (2.94)	70.19 (.06)	27.33 (.51)	59.52 (.26)	49.04 (.25)	50.19 (.7)	67.23 (.99)	54.94 (1.19)	91.37 (.16)	76.08 (.27)
BERT + MLCL (cos.)	**84.33** (.04)	60.52 (.06)	59.89 (.31)	49.33 (.42)	70.22 (.05)	30.42 (.56)	61.28 (.15)	51.41 (.32)	89.25 (1.36)	69.28 (1.89)	56.93 (1.3)	**91.87** (.21)	77.14 (.28)

Table 3: Macro-averaged F_1-scores on each hierarchical level.

ple, the score differences on most levels are as low as .05 F_1 points on the RCV1 dataset. However, McNemar tests show that the small differences between hyperbolic and non-hyperbolic LA models are statistically significant in the majority of the settings (cf. Table 7 in Appendix D). We observe that the hyperbolic models see more statistically significant performance increases on individual labels than decreases on labels in the majority of cases (e.g. 54 increases versus 36 decreases on the RCV1 dataset and 11 versus 7 on the WOS dataset). Most of these score increases are observed on the deeper leaf nodes.

Moreover, we also observe that the non-hyperbolic models generally achieve the best results in terms of hierarchical F_1 (cf. Table 2). Whereas hyperbolic exhibit the highest recall (especially with LA models), the non-hyperbolic counterparts excel at hierarchical precision.

These results indicate that the hyperbolic nature of the label embedding anchors are better suited to learn less-frequent deep leaf nodes because of the better separation of those anchors in the embedding space (Nickel and Kiela, 2017). However, there is a small trade-off between general performance across all classes and performance on the aforementioned more fine-grained labels on lower levels of the hierarchy, as shown by the higher F_1-micro and EMR yielded by the cosine LA losses. Additionally, the hyperbolic MLCL tend to underperform compared to non-hyperbolic counterparts, thus indicating that the hierarchical semantic relationships between similar texts can be better learned with loss functions in the Euclidean space. Therefore, these results question the overall effectiveness of the hyperbolic space for HMLTC.

These findings fall in line with the study from Fivez et al. (2021), which demonstrated that Euclidean anchors were equally or more effective than more complex anchors in the hyperbolic space for incorporating hierarchical semantic information into biomedical name representations. Additionally, our findings with regards to MLCL losses oppose some of the findings in the computer vision literature (Yue et al., 2023), which might be due to the different nature of the hierarchies and loss functions. Our results also differ from those in Chen et al. (2020a), who observed a superior per-

formance with hyperbolic representations of label-aware documents compared to Euclidean representations for HMLTC. This difference could stem from the different label representations and text encoders (GloVE + GRU), in that the language models that we use might capture hierarchical features in the Euclidean space better.

Label-aware vs. Contrastive Losses Comparing the LA models with the baselines, we observe that the LA models yield a decrease in true negatives, paired with an increase in true positives. The MLCL models on the other hand generally yield an increase in true negatives and true positives (Figure 4 - 7 in Appendix C).

Additionally, we observe that the Label-aware losses (LA) improve macro-averaged F_1 scores up to 7 points, of which the most substantial improvements are achieved when the Poincaré distance or Euclidean distance are used. Conversely, using the cosine distance as measure results in a lower increase in macro F_1 and a higher gain in micro F_1 and EMR compared to the hyperbolic LA loss. In general, the most considerable improvements with LA losses compared to the baselines are observed on the dataset with the most complex label hierarchies, namely BGC and RCV1-V2.

The MLCL models consistently show only a slight increase in macro F_1 ($\pm$ 2 F_1 points), but also an increase in micro F_1 and EMR across all datasets. Though the increases of these models are relatively low in terms of macro F_1 and micro F_1, they show more substantial improvements in consistency (EMR) compared to the baseline and LA models. The contrastive losses are the most effective when cosine distance is used as a distance measure.

The high increase in macro F_1 scores indicates that the models trained with the LA losses excel at predicting more infrequent and low-level classes, while the low increase in micro F_1 and EMR indicates that this is at the cost of the performance on some more frequent, high-level labels and overall consistency. This contrasts with MLCL-based models, where the increase in micro-averaged performance is relatively higher than the LA-based models. Across all datasets, we observe increases in performance on the lower frequent classes and low-level leaf nodes when using the LA models, with some classes seeing an increase of up to 80 F_1 points on the BGC dataset (for example, *Travel: Africa*).

Concerning the hierarchical consistency of the models, MLCL models generally achieve the best results in terms of hierarchical precision and F_1 (cf. Table 2), while the hyperbolic LA models exhibit the best hierarchical recall.

The reason why label-aware losses underperform compared to contrastive losses might lie in the nature of the label representations. The label-aware loss function, as adapted from Chen et al. (2023), does not use semantically rich label representations such as word2vec, GloVe or contextual embeddings as "base embeddings" to encode hierarchical relationships. Rather, these representations –as obtained from the graph neural network that encodes hierarchical relationships between nodes– are randomly initialized and merely encoded with the hierarchical positions. This might interfere with the rich contextual representations derived from BERT during training. The effectiveness of contrastive learning could be related to this explanation. We implicitly make use of the hierarchy by bringing instances of the same branches (i.e., instances with overlapping label sets) closer together in an embedding space. This does not introduce noise from an outside component (as is the case with label-aware losses), thereby showing more hierarchical consistency.

Error Analysis We further examine the differences between hyperbolic models and their non-hyperbolic counterparts by performing an error analysis. We aim to illustrate these differences by providing examples of the errors from the different models on the BGC dataset that we described in the previous paragraphs. The predicted label sets for several models are provided, with the hierarchical level for each node in between brackets.

Example a: LA (hyperb.) predicts an incorrect parent node

Text: "Luis Negrón's debut collection reveals the intimate world of a small community in Puerto Rico joined [...]"

- Truth: Fiction (1)
- LA (hyperb.): Fiction (1), **Poetry (1)**
- LA (cos.): Fiction (1)

Here the hyperbolic LA model incorrectly predicts Poetry, a first-level node, thereby highlighting the trade-off between performance on fine-grained nodes and overall consistency.

Example b: LA (hyperb.) correctly predicts fine-grained leaf nodes

Text: "With the utterance of a single line—"Doctor

Livingstone, I presume?"—a remote meeting [...]"

- Truth: Nonfiction (1), Biography & Memoir (2), History (2), World History (3), African World History (4)
- LA (hyperb.): Nonfiction (1), Biography & Memoir (2), History (2), **World History (3)**, **African World History (4)**
- LA (cos.): Nonfiction (1), Biography & Memoir (2), History (2)

Here the hyperbolic LA model correctly predicts two leaf nodes which other models fail to predict, thus highlighting the effectiveness of using the hyperbolic space for predicting deep nodes.

Example c: MLCL (Cos) correctly predicts a label set

Text: "Book Four in the Lydia Strong Series. In the final installment of her Lydia Strong series, best-selling author [...]"

- Truth: Fiction (1), Mystery & Suspense (2), Suspense & Thriller (3)
- LA (hyperb.): Fiction (1), Mystery & Suspense (2), Crime Mysteries (3), Suspense & Thriller (3)
- LA (cos.): Fiction (1), Mystery & Suspense (2)
- MLCL (cos.): Fiction (1), Mystery & Suspense (2), Suspense & Thriller (3)

Here the cosine-based MLCL model correctly predicts an entire label set, whereas the hyperbolic label-aware model incorrectly predicts two deep-level nodes.

5 Conclusion

In this study, we compared the performance of loss functions deployed in the Euclidean space and hyperbolic space. We noticed that the performance between hyperbolic and non-hyperbolic counterparts was generally the same. We also observed that incorporating the hyperbolic space either negatively affects or barely affects contrastive learning on most datasets and that cosine-based MLCL overall yields the best performance.

Label-aware losses generally performed well on fine-grained leaf nodes, though there was a trade-off between performance on fine-grained leaf nodes and overall performance or consistency depending on the metric and space. Additionally, we found that MLCL yielded an increased hierarchical performance and consistency overall, though these loss functions underperformed on fine-grained leaf nodes compared to the LA losses in most cases.

In summary, we found that Euclidean models yield a similar or even superior performance to hyperbolic models for HMLTC, challenging the efficacy of complex geometric embeddings given their marginal performance gains. Our work also highlights the challenge of optimally organizing hierarchical embeddings in a space, since the performance increases are rather small compared to more complex state-of-the-art approaches. Our study therefore takes a critical step in reevaluating the usage of different geometric space in NLP, which could inspire other similar studies in other AI applications.

6 Limitations

Previous work explored Supervised Contrastive Loss for (hierarchical) multi-label text classification, where transformations of a positive sample are leveraged as positive samples, as opposed to our presented MLCL models. Methods that generate additional positive samples could be more effective than our proposed MLCL functions and should be explored in the context of hyperbolic learning. Additionally, we leave it to future work to explore other hyperbolic distance measures for the proposed LA and MLCL models. In this work, we found differences between metrics in the Euclidean space, so we have to take into account that another hyperbolic distance metric could potentially yield different results. However, the limited performance increases on all datasets questions the general efficacy of additional training objectives for fine-tuning transformers on HMLTC datasets. With the advent of autoregressive LLMs like the GPT-models, it leaves the question how these models can be optimally prompted to perform HMLTC effectively.

In summary, exploring different MLCL functions (or other loss functions) and exploring metrics for measuring distances in the hyperbolic space should be valuable directions for future research.

Acknowledgments

This research was funded by the Flemish government under FWO IRI project CLARIAH-VL and the Flanders AI Research program.

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.

Alexandre Audibert, Aurélien Gauffre, and Massih-Reza Amini. 2024. Exploring contrastive learning for long-tailed multi-label text classification. *arXiv preprint arXiv:2404.08720*.

Soumya Chatterjee, Ayush Maheshwari, Ganesh Ramakrishnan, and Saketha Nath Jagarlapudi. 2021. Joint learning of hyperbolic label embeddings for hierarchical multi-label classification. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2829–2841, Online. Association for Computational Linguistics.

Boli Chen, Yao Fu, Guangwei Xu, Pengjun Xie, Chuanqi Tan, Mosha Chen, and Liping Jing. 2021a. Probing bert in hyperbolic spaces. *Preprint*, arXiv:2104.03869.

Boli Chen, Xin Huang, Lin Xiao, Zixin Cai, and Liping Jing. 2020a. Hyperbolic interaction model for hierarchical multi-label classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7496–7503.

Chih Yao Chen, Tun Min Hung, Yi-Li Hsu, and Lun-Wei Ku. 2023. Label-aware hyperbolic embeddings for fine-grained emotion classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10947–10958, Toronto, Canada. Association for Computational Linguistics.

Haibin Chen, Qianli Ma, Zhenxi Lin, and Jiangyue Yan. 2021b. 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.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020b. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Zhijian Chen, Zhonghua Li, Jianxin Yang, and Ye Qi. 2024. Hilight: A hierarchy-aware light global model with hierarchical local contrastive learning. *Preprint*, arXiv:2408.05786.

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.

Ankit Dhall, Anastasia Makarova, Octavian Ganea, Dario Pavllo, Michael Greeff, and Andreas Krause. 2020. Hierarchical image classification using entailment cone embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 836–837.

Pieter Fivez, Simon Suster, and Walter Daelemans. 2021. Integrating higher-level semantics into robust biomedical name representations. In *Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis*, pages 49–58, online. Association for Computational Linguistics.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic entailment cones for learning hierarchical embeddings. *CoRR*, abs/1804.01882.

Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, and Nando de Freitas. 2018. Hyperbolic attention networks. *Preprint*, arXiv:1805.09786.

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.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *CoRR*, abs/2004.11362.

Svetlana Kiritchenko, Stan Matwin, A Fazel Famili, et al. 2005. Functional annotation of genes using hierarchical text categorization. In *Proc. of the ACL Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*.

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.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397.

Nankai Lin, Guanqiu Qin, Gang Wang, Dong Zhou, and Aimin Yang. 2023. An effective deployment of contrastive learning in multi-label text classification. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8730–8744, Toronto, Canada. Association for Computational Linguistics.

Junyu Lu, Hao Zhang, Zhexu Shen, Kaiyuan Shi, Liang Yang, Bo Xu, Shaowu Zhang, and Hongfei Lin. 2022. Multi-task hierarchical cross-attention network for multi-label text classification. In *Natural Language Processing and Chinese Computing: 11th CCF International Conference, NLPCC 2022, Guilin, China, September 24–25, 2022, Proceedings, Part II*, page 156–167, Berlin, Heidelberg. Springer-Verlag.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Yue Wang, Dan Qiao, Juntao Li, Jinxiong Chang, Qishen Zhang, Zhongyi Liu, Guannan Zhang, and Min Zhang. 2023. Towards better hierarchical text classification with data generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7722–7739, Toronto, Canada. Association for Computational Linguistics.

Zihan Wang, Peiyi Wang, Lianzhe Huang, Xin Sun, and Houfeng Wang. 2022. 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.

Bo Xiong, Michael Cochez, Mojtaba Nayyeri, and Steffen Staab. 2022. Hyperbolic embedding inference for structured multi-label prediction. In *Advances in Neural Information Processing Systems*, volume 35, pages 33016–33028. Curran Associates, Inc.

Linli Xu, Sijie Teng, Ruoyu Zhao, Junliang Guo, Chi Xiao, Deqiang Jiang, and Bo Ren. 2021. Hierarchical multi-label text classification with horizontal and vertical category correlations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2459–2468, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. SGM: Sequence generation model for multi-label classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3915–3926, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Simon Chi Lok Yu, Jie He, Victor Basulto, and Jeff Pan. 2023. Instances and labels: Hierarchy-aware joint supervised contrastive learning for hierarchical multi-label text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8858–8875, Singapore. Association for Computational Linguistics.

Yun Yue, Fangzhou Lin, Kazunori D Yamada, and Ziming Zhang. 2023. Hyperbolic contrastive learning. *Preprint*, arXiv:2302.01409.

Ximing Zhang, Qian-Wen Zhang, Zhao Yan, Ruifang Liu, and Yunbo Cao. 2021. Enhancing label correlation feedback in multi-label text classification via multi-task learning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1190–1200, Online. Association for Computational Linguistics.

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

Zhenyu Zhang, Yuming Zhao, Meng Chen, and Xiaodong He. 2022b. Label anchored contrastive learning for language understanding. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1437–1449, Seattle, United States. Association for Computational Linguistics.

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.

Appendix

A Model Hyperparameters and Implementation Details

The table belows contain the hyperparameters used for the models.

Method	BGC	RCV1-V2	AAPD	WOS
BERT	LR = 5e-5	LR = 5e-5	LR = 2e-5	LR = 5e-5
BERT + LA (hyperb.)	LR = 5e-5 $\alpha = 2$	LR = 5e-5 $\alpha = 2$	LR = 5e-5 $\alpha = 1$	LR = 5e-5 $\alpha = 1$
BERT + LA (eucl.)	LR = 5e-5 $\alpha = 2$	LR = 5e-5 $\alpha = 1$	LR = 2e-5 $\alpha = 2$	LR = 5e-5 $\alpha = 1$
BERT + LA (cos.)	LR = 5e-5 $\alpha = 5$	LR = 5e-5 $\alpha = 5$	LR = 5e-5 $\alpha = 5$	LR = 5e-5 $\alpha = 1$
BERT + MLCL (hyperb.)	LR = 5e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.5$	LR = 5e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.5$	LR = 2e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.5$	LR = 5e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.5$
BERT + MLCL (Eucl.)	LR = 5e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.5$	LR = 7e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 1.0$	LR = 2e-5 $\alpha = 0.5$ JI = 1.0 $\theta = 0.5$	LR = 5e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.5$
BERT + MLCL (cos.)	LR = 5e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.5$	LR = 5e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.5$	LR = 2e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.7$	LR = 5e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.5$

Table 4: Hyperparameters for the BERT models. LR = Learning rate, α = alpha value for loss functions, JI = Jaccard Index, θ = margin.

Method	BGC	RCV1-V2	AAPD	WOS
DistilBERT	LR = 5e-5	LR = 5e-5	LR = 2e-5	LR = 5e-5
DistilBERT + LA (hyperb.)	LR = 5e-5 $\alpha = 2$	LR = 5e-5 $\alpha = 2$	LR = 5e-5 $\alpha = 1$	LR = 5e-5 $\alpha = 1$
DistilBERT + LA (eucl.)	LR = 5e-5 $\alpha = 2$	LR = 5e-5 $\alpha = 2$	LR = 5e-5 $\alpha = 1$	LR = 5e-5 $\alpha = 1$
DistilBERT + LA (cos.)	LR = 5e-5 $\alpha = 5$	LR = 5e-5 $\alpha = 5$	LR = 5e-5 $\alpha = 5$	LR = 5e-5 $\alpha = 1$
DistilBERT + CoLo (hyperb.)	LR = 5e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.5$	LR = 5e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.5$	LR = 5e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.5$	LR = 5e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.5$
DistilBERT + CoLo (Eucl.)	LR = 5e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.5$	LR = 7e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.7$	LR = 2e-5 $\alpha = 0.5$ J = 0.5 $\theta = 0.5$	LR = 7e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.7$
DistilBERT + CoLo (cos.)	LR = 5e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.5$	LR = 5e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.5$	LR = 2e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.5$	LR = 5e-5 $\alpha = 0.5$ JI = 0.5 $\theta = 0.5$

Table 5: Hyperparameters for the DistilBERT models. LR = Learning rate, α = alpha value for loss functions, JI = Jaccard Index, θ = margin.

The experiments were conducted on NVIDIA GeForce RTX 2080 Ti GPUs with 11,264GB of RAM. Each experiment took approximately 8 hours per random seed with BERT (110M parameters) and approximately 5 hours with DistilBERT (66M parameters). Training times varied slightly between datasets, though it should also be noted that the proposed approaches (label-aware and contrastive losses) do not add a substantial amount of training time compared to baselines ($\pm$5 minutes per epoch). The total computational cost of the experiments is roughly 650 GPU hours, including hyperparameter tuning experiments.

The implementations of precision, recall, F_1 and EMR in scikit-learn 1.2.0 were used to evaluate the models (Pedregosa et al., 2011).

B Results from DistilBERT

The tables below contain the results from the DistilBERT models. As noted in Section 4, the same trends are observed with BERT. Consult Section 4 for an in-depth discussion about the models' performance on the datasets.

	BGC			**RCV1-V2**			AAPD			**WOS**		
Setup	maF_1	miF_1	EMR	maF_1	miF_1	EMR	maF_1	miF_1	EMR	maF_1	miF_1	EMR
DistilBERT	59.22 (.5)	78.45 (.06)	45.46 (.14)	52.33 (.6)	78.88 (.27)	54.2 (.5)	58.32 (.74)	80.81 (.62)	39.16 (.83)	76.33 (.44)	86.41 (.28)	77.57 (.33)
DistilBERT + LA (hyperb.)	**62.99** (.29)	78.7 (.13)	44.87 (.23)	**59.09** (.24)	79.12 (.17)	53.87 (.36)	**58.92** (.91)	80.18 (.55)	37.36 (1.35)	**77.72** (.36)	86.56 (.13)	77.62 (.19)
DistilBERT + LA (eucl.)	62.54 (.43)	78.54 (.09)	44.45 (.15)	58.56 (.33)	78.99 (.2)	53.8 (.31)	58.89 (.92)	80.25 (.4)	37.44 (1.35)	76.98 (.18)	86.33 (.09)	76.96 (.1)
DistilBERT + LA (cos.)	61.9 (.51)	78.94 (.16)	45.9 (.14)	56.87 (.33)	79.64 (.11)	55.44 (.04)	58.48 (.81)	80.74 (.34)	38.46 (.34)	76.97 (.35)	86.39 (.26)	78.03 (.49)
DistilBERT + MLCL (hyperb.)	59.43 (.44)	78.56 (.16)	45.67 (.25)	52.68 (.54)	79.0 (.21)	54.58 (.27)	57.64 (.5)	80.72 (.5)	38.88 (1.46)	76.02 (.41)	86.27 (.15)	77.34 (.24)
DistilBERT + MLCL (Eucl.)	56.85 (.41)	78.66 (.12)	45.64 (.23)	51.16 (.34)	79.1 (.11)	54.17 (.34)	57.09 (.52)	80.92 (.35)	40.4 (.63)	76.83 (.15)	86.72 (.13)	78.2 (.19)
DistilBERT + MLCL (cos.)	60.61 (.27)	**79.43** (.06)	**46.79** (.3)	53.73 (.13)	**79.78** (.05)	**55.67** (.08)	58.35 (1.19)	**81.06** (.52)	**39.68** (1.41)	77.27 (.12)	**87.09** (.09)	**78.3** (.2)

Table 6: Results from DistilBERT. The best results across models are marked in bold.

C Confusion Matrices

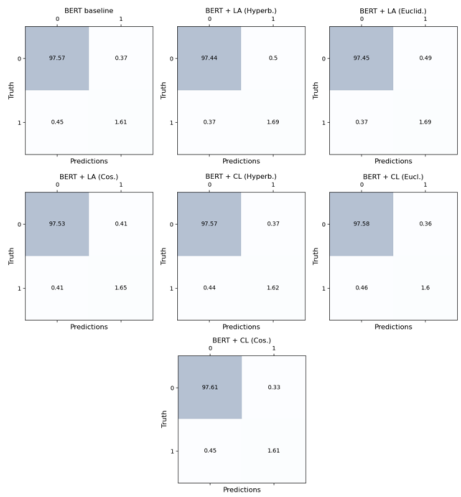

Figure 4: True negatives, false positives, false negatives and true positives (in %) from each model on the BGC dataset.

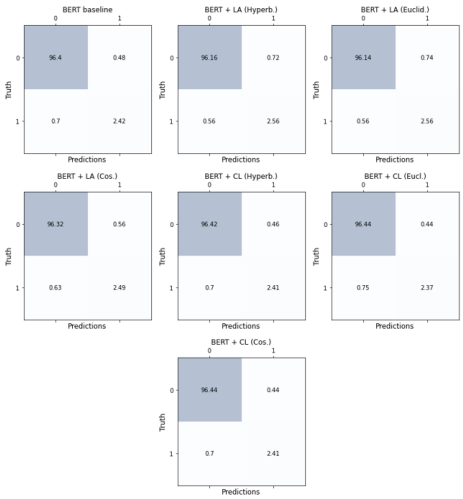

Figure 5: True negatives, false positives, false negatives and true positives (in %) from each model on the RCV1 dataset.

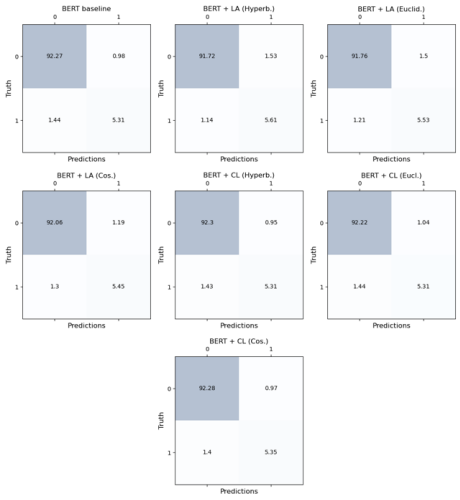

Figure 6: True negatives, false positives, false negatives and true positives (in %) from each model on the BGC dataset.

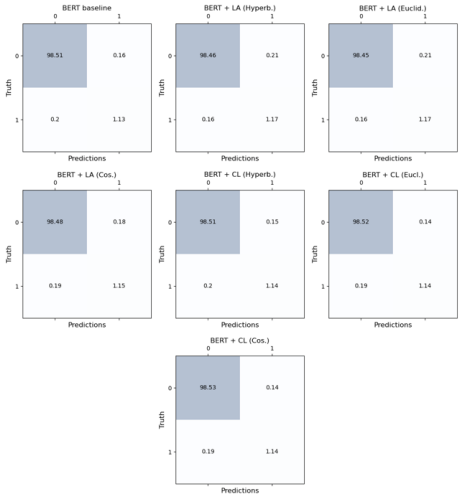

Figure 7: True negatives, false positives, false negatives and true positives (in %) from each model on the RCV1 dataset.

D Statistical Tests

Dataset	Setup	P-value	Improved Results (a > b)	Worse Results (a < b)
BGC	LA (hyperb.) vs. LA (eucl..)	.002	14 (9.5%) (1, 8, 3, 2)	14 (9.5%) (2, 3, 8, 1)
	LA (hyperb.) vs. LA (cos.)	<.001	64 (43.8%) (3, 18, 30, 13)	51 (28.8%) (3, 19, 28, 1)
RCV1	LA (hyperb.) vs. LA (eucl.)	<.001	54 (52.4%) (4, 14, 17, 18, 1)	36 (34.9%) (0, 5, 13, 18, 0)
	LA (hyperb.) vs. LA (cos.)	<.001	46 (44.6%) (3, 13, 15, 14, 1)	51 (49.5%) (0, 7, 18, 26, 0)
AAPD	LA (hyperb.) vs. LA (eucl.)	<.001	6 (9.8%) (0, 6)	1 (1.6%) (0, 1)
	LA (hyperb.) vs. LA (cos.)	<.001	13 (21.3%) (1, 12)	12 (19.7%) (0, 12)
WOS	LA (hyperb.) vs. LA (eucl.)	.002	11 (7.3%) (0, 11)	7 (4.6%) (0, 7)
	LA (hyperb.) vs. LA (cos.)	<.001	37 (24.7%) (4, 33)	33 (22.0%) (1, 32)

Table 7: Results from the McNemar tests between model set-ups. The last two columns show statistically significant improvements and decreases per hierarchical level.