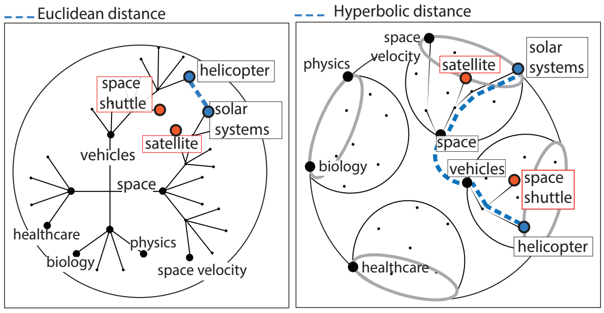
HyHTM: Hyperbolic Geometry based Hierarchical Topic Models

Simra Shahid [*] **Tanay Anand**[*] **Nikitha Srikanth**[*]
Sumit Bhatia **Balaji Krishnamurthy** **Nikaash Puri**
Media and Data Science Research Lab, Adobe, India
`{sshahid, tana, srikanth, sumit.bhatia, kbalaji, nikpuri}@adobe.com`

Abstract

Hierarchical Topic Models (HTMs) are useful for discovering topic hierarchies in a collection of documents. However, traditional HTMs often produce hierarchies where lower-level topics are unrelated and not specific enough to their higher-level topics. Additionally, these methods can be computationally expensive. We present **HyHTM** - a **Hy**perbolic geometry based **H**ierarchical **T**opic **M**odels - that addresses these limitations by incorporating hierarchical information from hyperbolic geometry to explicitly model hierarchies in topic models. Experimental results with four baselines show that HyHTM can better attend to parent-child relationships among topics. HyHTM produces coherent topic hierarchies that specialise in granularity from generic higher-level topics to specific lower-level topics. Further, our model is significantly faster and leaves a much smaller memory footprint than our best-performing baseline. We have made the source code for our algorithm publicly accessible. [1]



(a) Crowding of words in Euclidean spaces

(b) No crowding of words in Hyperbolic spaces

Figure 1: In figure (a) we see a concept tree in Euclidean spaces. Words such as *space shuttle* and *satellite*, which belong to moderately different super-concepts such as *vehicles* and *space*, respectively, are brought closer together due to their semantic similarity. This leads to a convergence of their surrounding words, such as *helicopter* and *solar system*, creating a false distance relationship and a crowding effect in Euclidean spaces. In figure (b), we see a concept tree in Hyperbolic spaces (Poincaré ball), which inherently has more space (represented by grey circles) than Euclidean spaces. The distances here grow exponentially towards the edge of the ball, and the concepts at deeper levels such as *helicopter* and *solar systems* move apart in these growing spaces and are far from each other. The dashed blue line shows how the distances in both spaces are calculated.

1 Introduction

The topic model family of techniques is designed to solve the problem of discovering human-understandable topics from unstructured corpora (Paul and Dredze, 2014) where a topic can be interpreted as a probability distribution over words (Blei et al., 2001). Hierarchical Topic Models (HTMs), in addition, organize the discovered topics in a hierarchy, allowing them to be compared with each other. The topics at higher levels are generic and broad while the topics lower down in the hierarchy are more specific (Teh et al., 2004).

While significant efforts have been made to develop HTMs (Blei et al., 2003; Chirkova and Vorontsov, 2016; Isonuma et al., 2020; Viegas et al., 2020), there are still certain areas of improvement. First, the ordering of topics generated by these approaches provides little to no information about the granularity of concepts within the corpus. By granularity, we mean that topics near the root should be more generic, while topics near the leaves should be more specific. Second, the lower-level topics must be related to the corresponding higher-level topics. Finally, some of these approaches such as CluHTM (Viegas et al., 2020) are very computationally intensive. We argue that these HTMs have such shortcomings primarily because they do not explicitly account for the hierarchy of words between topics.

Most of the existing approaches use document representations that employ word embeddings from euclidean spaces. These spaces tend to suffer from the **crowding problem** which is the tendency to accommodate moderately distant words close to each

[*] Authors contributed equally to the work.

[1] Our code is released at: `https://github.com/simra-shahid/hyhtm`

Findings of the Association for Computational Linguistics: ACL 2023, pages 11672–11688
July 9-14, 2023 ©2023 Association for Computational Linguistics

other (Van der Maaten and Hinton, 2008). There are several notable efforts that have shown that Euclidean spaces are suboptimal for embedding concepts in hierarchies such as trees, words, or graph entities (Chami et al., 2019, 2020; Guo et al., 2022). In figure 1(a), we show the crowding of concepts in euclidean spaces. Words such as space shuttle and satellite, which belong to moderately different concepts such as vehicles and space, respectively, are brought closer together due to their semantic similarity. This also leads to a convergence of their surrounding words, such as helicopter and solar system creating a false distance relationship. As a result of this crowding, topic models such as CluHTM that use Euclidean word similarities in their formulation tend to mix words that belong to different topics.

Contrary to this, hyperbolic spaces are naturally equipped to embed hierarchies with arbitrarily low distortion (Nickel and Kiela, 2017; Tifrea et al., 2019; Chami et al., 2020). The way distances are computed in these spaces are similar to tree distances, i.e., children and their parents are close to each other, but leaf nodes in completely different branches of the tree are very far apart (Chami et al., 2019). In figure 1(b), we visualise this intuition on a Poincaré ball representation of hyperbolic geometry (discussed in detail in Section 3). As a result of this tree-like distance computation, hyperbolic spaces do not suffer from the crowding effect and words like helicopter and satellite are far apart in the embedding space.

Inspired by the above intuition and to tackle the shortcomings of traditional HTMs, we present **HyHTM**, a **Hy**perbolic geometry based **H**ierarchical **T**opic **M**odel which uses hyperbolic geometry to create topic hierarchies that better capture hierarchical relationships in real-world concepts. To achieve this, we propose a novel method of incorporating semantic hierarchy among words from hyperbolic spaces and encoding it explicitly into topic models. This encourages the topic model to attend to parent-child relationships between topics.

Experimental results and qualitative examples show that incorporating hierarchical information guides the lower-level topics and produces coherent, specialised, and diverse topic hierarchies (Section 6). Further, we conduct ablation studies with different variants of our model to highlight the importance of using hyperbolic embeddings for representing documents and guiding topic hierarchies (Section 7). We also compare the scalability of our model with different sizes of datasets and find that our model is significantly faster and leaves much smaller memory footprint than our best-performing baseline (Section 6.1). We also present qualitative results in Section 6.2), where we observe that HyHTM topic hierarchies are much more related, diverse and specialised. Finally, we discuss and perform in-depth ablations to show the role of hyperbolic spaces and importance of every choice we made in our algorithm (See Section 7).

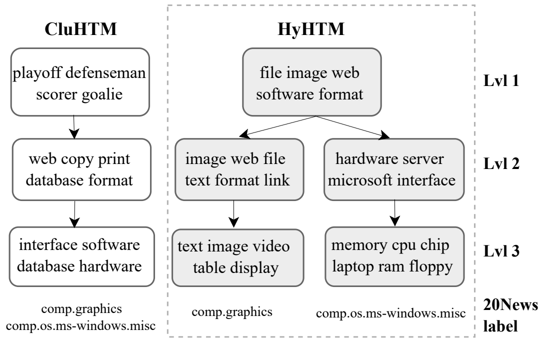

CluHTM
HyHTM
playoff defenseman scorer goalie
file image web software format
Lvl 1
web copy print database format
image web file text format link
hardware server microsoft interface
Lvl 2
interface software database hardware
text image video table display
memory cpu chip laptop ram floppy
Lvl 3
comp.graphics comp.os.ms-windows.misc
comp.graphics
comp.os.ms-windows.misc
20News label

Figure 2: Comparing our Hyperbolic-based HyHTM model to the Euclidean-based CluHTM, for selected 20News document labels (comp.graphics, comp.os.ms-windows.misc), we find that HyHTM is better at discriminating between similar document labels. CluHTM's root-level topics are not related to computer concepts, and it cannot separate these labels at lower levels. HyHTM groups them in the same root level and separates them into different lower-level topics, showing the advantage of using hyperbolic embeddings over euclidean ones to avoid the crowding problem. We show the top words with the highest probability for the topics.

2 Related Work

To the best of our knowledge, HTMs can be classified into three categories, **(I) Bayesian generative model** like hLDA (Blei et al., 2003), and its variants (Paisley et al., 2013; Kim et al., 2012; Tekumalla et al., 2015) utilize bayesian methods like Gibbs sampler for inferring latent topic hierarchy. These are not scalable due to the high computational requirements of posterior inference. **(II) Neural topic models** like TSNTM (Isonuma et al., 2020) and others (Wang et al., 2021; Pham and Le, 2021) use neural variational inference for faster parameter inference and some heuristics to learn topic hierarchies but lack the ability to learn appropriate semantic embeddings for topics. Along with these methods, there are **(III) Non-negative matrix factorization (NMF)** based topic models, which decompose a term-document matrix (like bag-of-words) into low-rank factor matrices to find latent topics. The hierarchy is learned using some heuristics (Liu et al., 2018a,b) or regularisation methods (Chirkova and Vorontsov, 2016) based on topics in the previous level.

However, the sparsity of the BoW representation for all these categories leads to incoherent topics, especially for short texts. To overcome this, some approaches have resorted to incorporating external knowledge from knowledge bases (KBs) (Duan et al., 2021b; Wang et al.) or leveraging word embeddings (Meng et al., 2020). Pre-trained word embeddings are trained on a large corpus of text data and capture the relationships between words such as semantic similarities, and concept hierarchies. These are used to guide the topic hierarchy learning process by providing a semantic structure to the topics. Viegas et al. (2020) utilizes euclidean embeddings for learning the topic hierarchy. However, Tifrea et al. (2019); Nickel and Kiela (2017); Chami et al. (2020); Dai et al. (2021) have shown how the crowding problem in Euclidean spaces makes such spaces suboptimal for representing word hierarchies. These works show how Hyperbolic spaces can model more complex relationships better while preserving structural properties like concept hierarchy between words. Recently, shi Xu et al. made an attempt to learn topics in hyperbolic embedding spaces. Contrary to the HTMs above, this approach adopts a bottom-up training where it learns topics at each layer individually starting from the bottom, and then during training leverages a topic-linking approach from Duan et al. (2021a), to link topics across levels. They also have a supervised variant that incorporates concept hierarchy from KBs.

Our approach uses latent word hierarchies from pretrained hyperbolic embeddings to learn the hierarchy of topics that are related, diverse, specialized, and coherent.

3 Preliminaries

We will first review the basics of Hyperbolic Geometry and define the terms used in the remainder of this section. We will then describe the basic building blocks for our proposed solution, followed by a detailed description of the underlying algorithm.

3.1 Hyperbolic Geometry

Hyperbolic geometry is a non-Euclidean geometry with a constant negative Gaussian curvature. Hyperbolic geometry does not satisfy the parallel postulate of Euclidean geometry. Consequently, given a line and a point not on it, there are at least two lines parallel to it. There are many models of hyperbolic geometry, and we direct the interested reader to an excellent exposition of the topic by Cannon et al. (1997). We base our approach on the **Poincaré ball** model, where all the points in the geometry are embedded inside an n-dimensional unit ball equipped with a metric tensor (Nickel and Kiela, 2017). Unlike Euclidean geometry, where the distance between two points is defined as the length of the line segment connecting the two points, given two points $u \in \mathbb{D}^n$ and $v \in \mathbb{D}^n$, the distance between them in the Poincaré model is defined as follows:

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

Here, arcosh is the inverse hyperbolic cosine function, and $\|.\|$ is the Euclidean norm. Figure 1 has shown an exemplary visualization of how words get embedded in hyperbolic spaces using the Poincaré ball model. As illustrated in Figure 1(b), distances in hyperbolic space follow a *tree-like* path, and hence they are informally also referred to as **tree distances**. As can be observed from the figure, the distances grow exponentially larger as we move toward the boundary of the Poincaré ball. This alleviates the crowding problem typical to Euclidean spaces, making hyperbolic spaces a natural choice for the hierarchical representation of data.

3.2 Matrix Factorization for Topic Models

A *topic* can be defined as a ranked list of strongly associated terms representative of the documents belonging to that topic. Let us consider a document corpus $\mathcal{D}$ consisting of n documents $d_1, d_2, \ldots, d_n$, and let $\mathcal{V}$ be the corpus vocabulary consisting of m distinct words $w_1, w_2, \ldots, w_m$. The corpus can also be represented by a document-term matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ such that $\mathbf{A}_{ij}$ represents the relative importance of word w_j in document d_i (typically represented by the TF-IDF weights of w_i in d_j).

A popular way of inferring topics from a given corpus is to factorize the document-term matrix. Typically, non-negative Matrix Factorization (NMF) is employed to decompose the document-term matrix, $\mathbf{A}$, into two non-negative approximate factors: $\mathbf{W} \in \mathbb{R}^{n \times \mathbf{N}}$ and $\mathbf{H} \in \mathbb{R}^{\mathbf{N} \times m}$. Here, $\mathbf{N}$ can be interpreted as the number of underlying topics. The factor matrix $\mathbf{W}$ can then be interpreted as the document-topic matrix, providing the topic memberships for documents, and **H**, the topic-term matrix, describes the probability of a term belonging to a given topic. This basic algorithm can also be applied recursively to obtain a hierarchy of topics by performing NMF on the set of documents belonging to each topic produced at a given level to get more fine-grained topics (Chirkova and Vorontsov, 2016; Viegas et al., 2020).

4 Hierarchical Topic Models Using Hyperbolic Geometry

We now describe HyHTM – our proposed Hyperbolic geometry-based Hierarchical Topic Model. We first describe how we capture semantic similarity and hierarchical relationships between terms in hyperbolic space. We then describe the step-by-step algorithm for utilizing this information to generate a topic hierarchy.

4.1 Learning Document Representations in Hyperbolic Space and Root Level Topics

As discussed in Section 3.2, the first step in inferring topics from a corpus using NMF is to compute the document-term matrix **A**. A typical way to compute the document-term matrix **A** is by using the TF-IDF weights of terms in a document that provides reprsentations of the documents in the term space. However, usage of TF-IDF (and its variants) results in sparse representations and ignores the semantic relations between different terms by considering only the terms explicitly present in a given document. Viegas et al. (2019) proposed an alternative formulation for document representations that utilizes pre-trained word embeddings to enrich the document representations by incorporating weights for words that are semantically similar to the words already present in the document. The resulting document representations are computed as follows.

$$\mathbf{A} = (\mathbf{TF} \times \mathbf{M}_\mathrm{S}) \odot (\mathbf{1} \times \mathbf{IDF}^T) \tag{2}$$

Here, $\odot$ indicates the Hadamard product. **A** is the $n \times m$ document-term matrix. **TF** is the term-frequency matrix such that $\mathbf{TF}_{i,j} = tf(d_i, w_j)$ and $\mathbf{M}_\mathrm{S}$ is the $m \times m$ term-term similarity matrix that captures the pairwise semantic relatedness between the terms and is defined as $\mathbf{M}_{\mathrm{S}_{i,j}} = sim(w_i, w_j)$, where $sim(w_i, w_j)$ represents the similarity between terms w_i and w_j and can be computed using typical word representations such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). Finally, **IDF** is the $m \times 1$ inverse-document-frequency vector representing the corpus-level importance of each term in the vocabulary. Note that Viegas et al. (2019) used the following modified variant of IDF in their formulation, which we also chose in this work.

$$\mathbf{IDF}(i) = \log\left(\frac{|D|}{\sum_{d\in D} \mu\left(w_i, d\right)}\right) \tag{3}$$

Here, $\mu\left(w_i, d\right)$ is the average of the similarities between term w_i and all the terms w in document d such that $\mathbf{M}_\mathrm{S}(w_i, w) \neq 0$. Thus, unlike traditional IDF formulation where the denominator is document-frequency of a term, the denominator in the above formulation captures the semantic contribution of w_i to all the documents.

In our work, we adapt the above formulation to obtain document representations in Hyperbolic spaces by using **Poincaré GloVe embeddings** (Tifrea et al., 2019), an extension of the traditional Euclidean space GloVe (Pennington et al., 2014) to hyperbolic spaces. Due to the nature of the Poincaré Ball model, the resulting embeddings in the hyperbolic space arrange the correspondings words in a hierarchy such that the sub-concept words are closer to their parent words than the sub-concept words of other parents.

There is one final missing piece of the puzzle before we can obtain suitable document representations in hyperbolic space. Recall that due to the nature of the Poincaré Ball model, despite all the points being embedded in a unit ball, the hyperbolic distances between points, i.e., tree distances (Section 3.1) grow exponentially as we move towards boundary of the ball (see Figure 1). Consequently, the distances are bounded between 0 and 1. As NMF requires all terms in the input matrix to be positive, we cannot directly use these distances to compute the term-term similarity matrix $\mathbf{M}_\mathrm{S}$ in Equation (2) as $1 - d_P\left(w, w'\right)$ can be negative. To overcome this limitation, we introduce the notion of **Poincaré Neighborhood Similarity**, (s_{p_n}), which uses a neighborhood normalization technique. The k-neighborhood of a term w is defined as the set of top k-nearest terms $w_1, ..., w_k$ in the hyperbolic space and is denoted as $n_k(w)$. For every term in the vocabulary $\mathcal{V}$, we first calculate the pair-wise poincaré distances with other terms using Equation (1). Then, for every term $w \in \mathcal{V}$, we compute similarity scores with all the other terms in its k-neighborhood $n_k(w)$ by dividing each pair-wise poincaré distance between the term and its neighbor by the maximum pair-wise distance in the neighborhood. This can be represented by the following equation where $w' \in n_k(w)$:

$$s_{p_n}\left(w, w'\right) = 1 - \frac{d_P\left(w, w'\right)}{\max\limits_{w_a, w_b \in n_k(w)}\left(d_P\left(w_a, w_b\right)\right)} \quad (4)$$

With this, we can now compute the term-term similarity matrix $\boldsymbol{M_S}$ as follows.

$$\mathbf{M}_{\mathrm{S}}(w, w') = \begin{cases} s_{p_n}\left(w, w'\right) & \text{if } s_{p_n}\left(w, w'\right) \geq \alpha, \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Note that there are two hyperparameters to control the neighborhood – *(i)* the neighborhood size using k_s; and *(ii)* the quality of words using α, which keeps weights only for the pair of terms where the similarity crosses the pre-defined threshold α thereby reducing noise in the matrix. Without α, words with very low similarity may get included in the neighborhood eventually leading to noisy topics.

We now have all the ingredients to compute the document-representation matrix $\mathbf{A}$ in the hyperbolic space and NMF can be performed to obtain the first set of topics from the corpus as described in Section 3.2. This gives us the *root* level topics of our hierarchy. Next, we describe how we can discover topics at subsequent levels.

4.2 Building the Topic Hierarchy

In order to build the topic hierarchy, we can iteratively apply NMF for topics discovered at each level as is typically done in most of the NMF based approaches. However, do note that working in the Hyperbolic space allows us to utilize hierarchical information encoded in the space to better guide the discovery of topic hierarchies. Observe that the notion of similarity in the hyperbolic space as defined in Equation(4) relies on the size of the neighborhood. In large neighborhood, a particular term will include not only its immediate children and ancestors but also other semantically similar words that may not be hierarchically related. On the other hand, a small neighborhood will include only the immediate parent-child relationships between the words, since subconcept words are close to their concept words. HyHTM uses this arrangement of words in hyperbolic space to explicitly guide the lower-level topics to be more related and specific to higher-level topics. In order to achieve this, we construct a **Term-Term Hierarchy** matrix, $\mathbf{M}_{\mathrm{H}} \in R^{|V| \times |V|}$ as follows.

$$\mathbf{M}_{\mathrm{H}}(w, w') = \begin{cases} 1 & \text{if } w' \in n_{k_h}(w), \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Here, k_H is a hyperparameter that controls the neighborhood size. $\mathbf{M}_{\mathrm{H}}$ is a crucial component of our algorithm as it encodes the hierarchy information and helps guide the lower-level topics to be related and specific to the higher-level topics.

Without loss of generality, let us assume we are at i^{th} topic node t_i at level l in the hierarchy. We begin by computing $\mathbf{A}_0 = \mathbf{A}$, as outlined in Equation (2), at the root node (representing all topics) and subsequently obtaining the first set of topics (at level $l = 1$). Also, let the number of topics at each node in the hierarchy be N (a user-specified parameter). Every document is then assigned to one topic with which it has the highest association in the document-topic matrix $\mathbf{W}_{l-1}$. Once all the documents are collected into disjoint parent topics, we use a subset of $\mathbf{A}_0$ with only the set of documents ($\mathcal{D}_{t_j}$) belonging to the j^{th} topic, and denote this by $\mathbf{A}_{l-1}$. We then branch out to N lower-level topics at the i^{th} node, using the following steps:

Parent-Child Re-weighting for Topics in the Next Level: We use the term-term hierarchical matrix $\mathbf{M}_{\mathrm{H}}$ to assign more attention to words hierarchically related to all the terms in the topic node t_i, and guide the topic hierarchy so that the lower-level topics are consistent with their parent topics. We take the product of the topic-term matrix of the t_i, denoted by, $\mathbf{H}_i$ with the hierarchy matrix $\mathbf{M}_{\mathrm{H}}$. This assigns weights with respect to associations in the topic-term matrix

$$\mathbf{M}_{ti} = \mathbb{1}_i^T \mathbf{H}_{l-1} \times \mathbf{M}_{\mathrm{H}} \quad (7)$$

Here, $\mathbb{1}_i$ is the one-hot vector for topic i, and $\mathbf{H}_{l-1}$ is the topic-term factor obtained by factorizing the document-representations $\mathbf{A}_{l-1}$ of the parent level.

Document representation for computing next level topics: We now compute the updated document representations for documents in topic node t_i that infuse semantic similarity between terms with hierarchical information as follows.

$$\mathbf{A}_l = \mathbf{A}_{l-1} \odot \mathbf{M}_{ti} \quad (8)$$

By using the updated document representations $\mathbf{A}_l$ we perform NMF as usual and obtain topics for level $l + 1$. The algorithm then continues to discover topics at subsequent levels and stops exploring topic hierarchy under two conditions – *(i)* if it reaches a topic node such that the number of documents in the node is less than a threshold (D_{min}); *(ii)* when the maximum hierarchy depth ($\mathcal{L}_{max}$) is reached. We summarize the whole process in the form of a pseudocode in Algorithm 1.

Algorithm 1: The HyHTM Algorithm

```
Input  : Max depth level ($\mathcal{L}_{\max}$)
         Min # of documents ($D_{\min}$)
         Default # of topics ($N$)
Output : Hierarchy of Topics
Compute $\mathbf{A}$ using Eq (2) & (5)
GetHier($\boldsymbol{A}$, 1)
def GetHier($\boldsymbol{A}$, $L$):
    if $L > L_{max}$ or len($\mathbf{A}$) $< D_{min}$:
      return
    $\mathbf{W}_{l-1}, \mathbf{H}_{l-1} \leftarrow$ NMF($\mathbf{A}, N$)
    for $i = 0$ to $H_{l-1}.size$ do
        Get parent topic using $\mathbf{H}_{l-1}$
        Add topic to hierarchy
        Get Docs of topic $t_j$ using $\mathbf{W}_{l-1}$
        Get $\mathbf{A}_{l-1}$ for $D_{t_j}$ from $\mathbf{A}_0$
        Compute Parent-Child
          Reweighting $\mathbf{M}_{ti}$ using Eq (7)
        Compute $\mathbf{A}_l$ next level from $\mathbf{M}_{ti}$
          & $\mathbf{A}_{l-1}$ using Eq (8)
        GetHier($\boldsymbol{A}_l$, $L + 1$)
```

5 Experimental Setup

Datasets: To evaluate our topic model, we consider 8 well-established public benchmark datasets. In Table 1 we report the number of words and documents, as well as the average number of words per document. We have used datasets with varying numbers of documents and average document lengths. We provide preprocessing details in the Appendix (See C.1).

Dataset	Vocabulary	No. of Documents	Avg. Doc Length
InfoVis-Vast (**InfoVAST**)	8,309	1,085	153.62
Neurips	9,407	1,499	517.9
BBC	6,384	2,255	209.00
20Newsgroup (**20News**)	12,199	18,846	119.80
Enron	10,116	39,860	93.29
Amazon Reviews (**Amazon**)	9,458	40,000	39.04
Web of Science (**WOS**)	40,755	46,985	132.30
AGNews	17,436	127,600	24.15

Table 1: Dataset characteristics

Baseline Methods: Our model is a parametric topic model which requires a fixed number of topics to be specified. This is different from non-parametric models, which automatically learn the number of topics during training. For the sake of completeness, we also compare our model to various non-parametric models such as **hLDA** (Blei et al., 2003) a bayesian generative model, and **TSNTM** (Isonuma et al., 2020) which uses neural variational inference. We also compare with NMF-based parametric models like **hARTM** (Chirkova and Vorontsov, 2016) which learns a topic hierarchy with a bag of words of documents and **CluHTM** (Viegas et al., 2020) which uses euclidean based pre-trained embeddings (Mikolov et al., 2017) to provide semantic similarity context to topic models. We provide the implementation details of these baselines in the Appendix (See C).

Number of topics: hARTM only allows fixing the total number of topics at a level and cannot specify the number of child topics for every parent topic. CluHTM, on the other hand, has a method to learn the optimal number of topics, but it is highly inefficient[2]. We use the same number of topics for fair comparison in hARTM, CluHTM, and HyHTM. We fix the number of topics for the top level as 10, with 10 sub-topics under each parent topic. The total number of topics at each level is 10, 100, and 1000. Non-parametric models hLDA and TSNTM learn the number of topics, and we report these numbers in the appendix (See E).

We select the best values for the hyperparameters k_H, k_S, and α by tuning them for the model with the best empirical results. We report these in the Appendix C.

6 Experimental Results

In this section we compare our model's performance on well-estabilished metrics to assess the coherence, specialisation, and diversity of topics. We present qualitative comparision for selected topics in Figure 2 and in Appendix 6.2. We discuss and perform ablations to show the role of hyperbolic spaces and effectiveness of our algorithm (See Appendix 7).

RQ1: Does HyHTM produce coherent topics? Topic coherence is a measure that can be used to determine how much the words within a topic co-occur in the corpus. The more the terms co-occur, the easier it is to understand the topic. We employ

[2] The training time of CluHTM 20News was approximately 32 hours, and for Amazon was approximately 22 hours. For every branch and level, it runs an empirical analysis for topics in ranges 5 and 20 and picks the topic number corresponding to the best coherence.

the widely used coherence measure from Aletras and Stevenson (2013) and report the average across the top 5 and 10 words for every topic in Table 2. We observe that for majority of the datasets, HyHTM consistently ranks at the top or second highest in terms of coherence. We also observe that for some cases hLDA and TSNTM, which have very few topics (See E) compared to HyHTM, have higher coherence values. To this end, we conclude that incorporating neighborhood properties of words from hyperbolic spaces can help topic models to produce topics that are comprehensible and coherent. Coherence is mathematically defined as,

$$\text{Coherence} = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \log \frac{P(w_i,w_j)}{P(w_i)P(w_j)}}{\binom{n}{2}} \quad (9)$$

where w_i and w_j are words in the topic, while $P(w_i, w_j)$ and $P(w_j)$ are the probabilities of co-occurrence of w_i and w_j and the of occurrence of w_j in the corpus respectively.

Dataset	hLDA	TSNTM	hARTM	CluHTM	**HyHTM**
InfoVAST	**0.061**	0.017	0.044	0.027	0.045
Neurips	0.066	0.133	0.084	0.226	**0.338**
BBC	0.232	0.248	**0.296**	0.181	0.235
20News	0.214	0.279	**0.325**	0.293	**0.325**
Enron	0.226	0.250	0.327	0.346	**0.365**
Amazon	0.127	0.097	**0.166**	0.124	0.158
WOS	0.024	**0.096**	0.025	0.010	0.052
AGNews	0.145	**0.209**	0.142	0.039	0.154

Table 2: Comparing topic coherence, where higher coherence is better. Bold represents the best-performing metric and underline represents the second-best metric.

RQ2: Does HyHTM produce related and diverse hierarchies? To assess the relationships between higher-level parent topics and lower-level child topics, we use two metrics: (i) hierarchical coherence, and (ii) hierarchical affinity.

Hierarchical Coherence: We build upon the coherence metric above to compute the coherence between parent topic words and child topic words. For every parent-topic and child-topic pair, we calculate the average across the top 5 words and top 10 words and report this in Table 3. We observe that HyHTM outperforms the baselines across datasets, and we attribute this result to our parent-child reweighting framework of incorporating the hierarchy of higher-level topics. In most cases, hLDA and TSNTM have very low hierarchical coherence because the topics generated by these models are often too generic across levels and contain multiple words from different concepts, whereas hARTM and CluHTM have reasonable scores and are often better than these. From this observation, we conclude that adding hierarchies from hyperbolic spaces to topic models produces a hierarchy where lower-level topics are related to higher-level topics. Hierarchical coherence is defined as,

$$\text{HCoherence} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \log \frac{P(w_i,w_j)}{P(w_i)P(w_j)}}{n^2} \quad (10)$$

where w_i and w_j represent words from the parent topic and child topic, while $P(w_i, w_j)$ and $P(w_j)$ are the probabilities of co-occurrence of w_i and w_j and the of occurrence of w_j in the corpus respectively.

Dataset	hLDA	TSNTM	hARTM	CluHTM	**HyHTM**
InfoVAST	0.011	0.018	0.007	0.011	**0.025**
Neurips	0.059	0.019	0.049	0.063	**0.296**
BBC	0.064	0.089	0.211	0.102	**0.221**
20News	0.031	0.049	0.133	0.127	**0.287**
Enron	0.023	0.068	0.139	0.107	**0.329**
Amazon	0.008	0.056	0.073	0.085	**0.123**
WOS	0.006	0.022	0.016	0.002	**0.045**
AGNews	0.017	0.018	0.046	0.071	**0.151**

Table 3: Comparing Hierarchical Coherence. Bold represents the best-performing metric and underline represents the second-best metric.

Hierarchical Affinity: We employ this metric from Isonuma et al. (2020) which considers the topics at levels 2 as parent topics and the topics at level 3 to compute (i) **child affinity**, and, (ii) **non-child affinity**. The respective affinities are measured by the average cosine similarity of topic-term distributions between parent & child and parent & non-child topics. [3] When child affinity is higher than non-child affinity, it implies (i) the topic hierarchy has a good diversity of topics, and, (ii) the parents are related to their children. We present the hierarchical affinities in figure 3.

We observe that HyHTM has the largest between child affinities across all the datasets. We also observe that the difference between child and non-child affinities is also larger than that for any other baseline. hLDA and TSNTM have very similar child and non-child affinities, which indicates how generic topics are across the hierarchy. In hARTM, we observe high child affinity and negligible non-

[3] Hierarchical Affinity metric is independent of the embedding space the models were they are trained on.

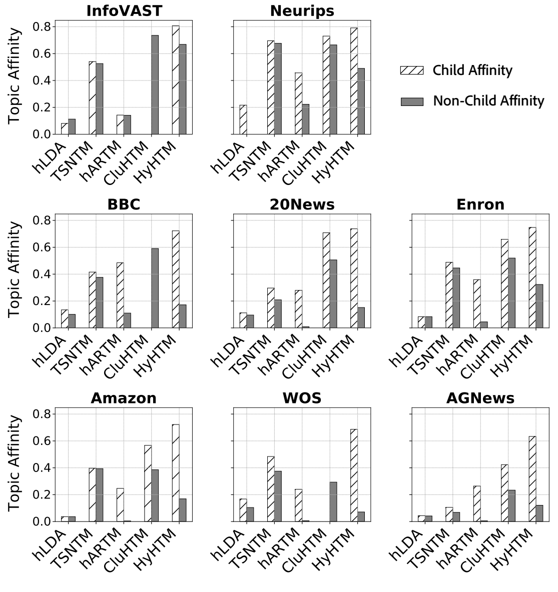


Figure 3: Analysis of Hierarchical Topic Affinities. A higher Child Affinity value indicates stronger relatedness between parent and child topics. The more the difference between Child to Non-Child Affinities, the more diverse the topics are in the hierarchy. Please note, some affinities appear to be missing in the visualization due to their significantly lower magnitudes compared to the highest affinity value."

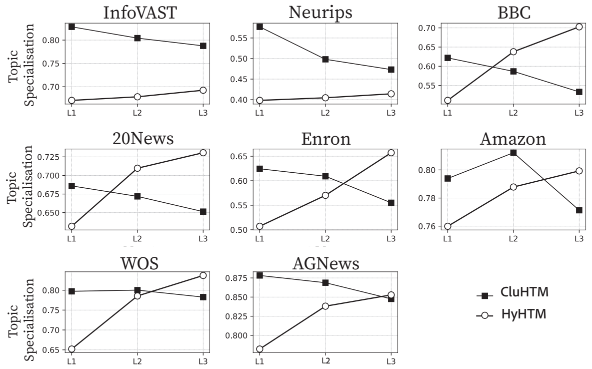


Figure 4: Comparision between Topic Specialisation of CluHTM and HyHTM for different datasets. An increasing trend from Level 1 (L1) to Level 3 (L3) indicates that topics are becoming more specific, diverging from a more generic corpus-word distribution.

child affinity. From these observations, we conclude that HyHTM produces related and diverse topics.

RQ3: Does HyHTM produce topics with varying granularity across levels? We use the Topic specialisation metric from Kim et al. (2012), to understand the granularity of topics in the hierarchy. Topic specialization is the cosine distance between the term distribution of a topic with the term distribution of the whole corpus. According to the metric, the root-level topics are trained on the whole corpus so they are very generic, while the lower-level topics are trained on a subset of documents, and they specialise. A higher specialization value means that the topic vector is not similar to the whole corpus vector, and hence it is more specialised. With increasing depth in the hierarchy, the specialisation of a topic should increase and its distance from the corpus vector must increase to model reasonable topic hierarchies described above.

As the resulting topic-proportions and range of topic-specialisation of CluHTM and HyHTM are similar, we first focus on these models to effectively underscore the advantages of employing hyperbolic spaces. As depicted in Figure 4, unlike CluHTM, our HyHTM model consistently exhibits an increasing trend in topic specialization across majority of the datasets. We attribute this result to our use of hyperbolic spaces in our algorithm which groups together documents of similar concepts from the root level itself.

Additionally, we present the topic specialization of other models in Appendix Table 5. We discover that TSNTM usually scores low, suggesting generic topics at all levels. Although hLDA shows increasing specialization, it seemingly fails to generate related topic hierarchies, as evidenced by quantitative metrics and qualitative topics (See Section 6.2). Despite hARTM showing an increase in granularity, it often lumps unrelated concepts under a single topic hierarchy, akin to CluHTM, as illustrated in the qualitative examples (See Section 6.2).

6.1 Runtime & Memory footprint

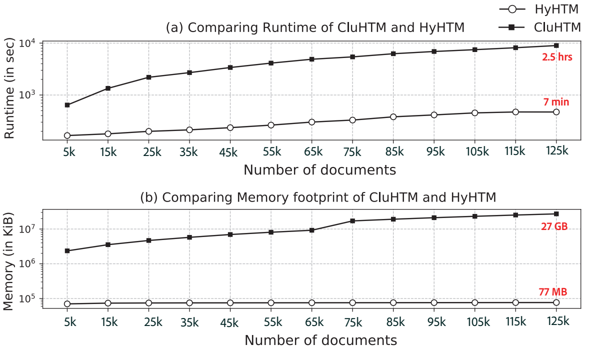


Figure 5: Comparing runtime and memory footprint for HyHTM (our model) and CluHTM on AGNews dataset.

To evaluate how our model scales with the size of the datasets, we measure the training time and memory footprint by randomly sampling a different number of documents (5k to 125k) from the AGNews dataset. From Figure 5 we observe that, as the number of documents increases, the training time of our model does not change considerably, whereas that of the CluHTM increases significantly. HyHTM can be trained approximately 15 times faster than the CluHTM model with even 125k documents. CluHTM works inefficiently by keeping the document representations of all the topics at a level in the working memory. This is a result of CluHTM developing the topic hierarchy in a breadth-first manner. We have optimized the HyHTM code to train one branch from root to leaf in a depth-first manner which makes our model more memory and efficient. hLDA took approximately 1.32 hours for training on the complete dataset, and hARTM and TSNTM took more than 6 hours.

6.2 Quality of Topics

To intuitively demonstrate the ability of our model to generate better hierarchies, we present topic hierarchies of all models for some selected 20News target labels in the Appendix in Figure 6. [4] Across various topic categories, unlike HyHTM, other models tend to struggle with delineating specific subconcepts, maintaining relatedness, and ensuring specialization within their topics, which highlights HyHTM's improved comprehensibility. For the *sci.space* 20News label, we observe that topics from CluHTM across all the levels are related space concepts but it is challenging to label them as specific subconcepts. The hARTM topics for space has a resonable hierarchy but it has documents of different concepts such as *sci.space, sci.med, rec.sports.baseball*. For hLDA and TSNTM, the lack of relatedness and specialization makes it difficult to identify these topics as space-themed. A similar trend can be observed for *comp.os.ms-windows.misc* and *sci.med* 20News categories in the figure, where the models exhibit similar struggles.

7 Ablation

Do Hyperbolic embeddings represent documents better than Euclidean ones?
To investigate this we consider a variant of our model called **Ours (Euc)** which incorporates pretrained Fasttext (Bojanowski et al., 2017) (trained on euclidean spaces) instead of Poincare embeddings in $M_s(w, w')$, and we keep all the other steps unchanged. From Table 4, we observe that using hyperbolic embeddings for guiding parent-child in A_l is better choice as it produces topics that are more coherent and hierarchies in which lower-level topics are related to higher-level topics.

	20News		Amazon	
	Coh	Hier Coh	Coh	Hier Coh
Ours	**0.325**	**0.287**	**0.158**	**0.123**
Ours (Euc)	0.322	0.240	0.156	0.113
CluHTM	0.293	0.127	0.124	0.085

Table 4: Analysis the role of hyperbolic embeddings

Does enforcing hierarchy between parent-child topics in equation 8 result in better hierarchy?
We examine this by comparing the **Ours (Euc)** variant and the CluHTM baseline. Both models use identical underlying document representations, yet they differ in how they guide their hierarchies, particularly in the equation 8 of our model. As demonstrated in Table 4, **Ours (Euc)**, which accounts for word hierarchies between higher-level and lower-level topics, generates topic hierarchies that are nearly twice as effective in terms of topical hierarchical coherence and hierarchical affinity.

In the Appendix (See Section B), we also examine the importance of our approach by replacing the underlying algorithm with hierarchical clustering methods.

8 Conclusion

In this paper, we have proposed HyHTM, which uses hyperbolic spaces to distill word hierarchies of higher-level topics in order to refine lower-level topics. Both quantitative and qualitative experiments have demonstrated the effectiveness of HyHTM in creating hierarchies in which lower-level topics are realted and more specific than higher-level topics. HyHTM is much more efficient compared to our best-performing baseline. A major limitation of HyHTM is that it is parametric and therefore requires empirical analysis to find the optimal number of topics at each level. We plan to investigate this shortcoming in the future.

[4] We present only those topic-hierarchies where most of the documents of the respective 20News label lies.

9 Limitations

In this paper, we propose a method to effectively incorporate the inherent word hierarchy in topic models for hierarchical topic mining. We use poincare embeddings, trained on wikipedia, to compute the hierarchical relatedness between words. Hence, our model relies on how well these embeddings are trained and whether they effectively capture the word hierarchy. Moreover, any bias in the embeddings is translated into our model. The second major limitation of our model is that since these embeddings are trained on wikipedia, they may not perform well on datasets that are very different from wikipedia or on datasets where the relation between two words is very different from their relation in wikipedia. For example, *topic* and *hierarchy* will have a very different relation in scientific journals from what they have in wikipedia. Our model is parametric HTM, and we plan on investigating methods to induce number of topics using hyperbolic spaces.

10 Ethics Statement

- The dataset used to train the poincare embeddings is Wikipedia Corpus, a publicly available dataset standardized for research works.
- We have added references for all the papers, open-source code repositories and datasets.
- In terms of dataset usage for topic modeling, we have used only publicly available datasets. We also ensure that any datasets used in our research do not perpetuate any harmful biases.
- We also plan to make our models publicly available, in order to promote transparency and collaboration in the field of natural language processing.

References

Nikolaos Aletras and Mark Stevenson. 2013. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 13–22, Potsdam, Germany. Association for Computational Linguistics.

David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. 2003. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pages 17–24. MIT Press.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2001. Latent dirichlet allocation. In *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pages 601–608. MIT Press.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

James W Cannon, William J Floyd, Richard Kenyon, Walter R Parry, et al. 1997. Hyperbolic geometry. *Flavors of geometry*, 31(59-115):2.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6901–6914, Online. Association for Computational Linguistics.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. 2019. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32.

NA Chirkova and KV Vorontsov. 2016. Additive regularization for hierarchical multimodal topic modeling. *Journal of Machine Learning and Data Analysis*, 2(2):187–200.

Shuyang Dai, Zhe Gan, Yu Cheng, Chenyang Tao, Lawrence Carin, and Jingjing Liu. 2021. APo-VAE: Text generation in hyperbolic space. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 416–431, Online. Association for Computational Linguistics.

Zhibin Duan, Dongsheng Wang, Bo Chen, Chaojie Wang, Wenchao Chen, Yewen Li, Jie Ren, and Mingyuan Zhou. 2021a. Sawtooth factorial topic embeddings guided gamma belief network. In *International Conference on Machine Learning*, pages 2903–2913. PMLR.

Zhibin Duan, Yi Xu, Bo Chen, Chaojie Wang, Mingyuan Zhou, et al. 2021b. Topicnet: Semantic graph-guided topic discovery. *Advances in Neural Information Processing Systems*, 34:547–559.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Yunhui Guo, Haoran Guo, and Stella X Yu. 2022. Co-sne: Dimensionality reduction and visualization for hyperbolic data. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 21–30.

Masaru Isonuma, Junichiro Mori, Danushka Bollegala, and Ichiro Sakata. 2020. Tree-Structured Neural Topic Model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 800–806, Online. Association for Computational Linguistics.

Joon Hee Kim, Dongwoo Kim, Suin Kim, and Alice Oh. 2012. Modeling topic hierarchies with the recursive chinese restaurant process. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 783–792.

Rui Liu, Xingguang Wang, Deqing Wang, Yuan Zuo, He Zhang, and Xianzhu Zheng. 2018a. Topic splitting: A hierarchical topic model based on non-negative matrix factorization. *Journal of Systems Science and Systems Engineering*, 27.

Rui Liu, Xingguang Wang, Deqing Wang, Yuan Zuo, He Zhang, and Xianzhu Zheng. 2018b. Topic splitting: a hierarchical topic model based on non-negative matrix factorization. *Journal of Systems Science and Systems Engineering*, 27(4):479–496.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Yu Zhang, Chao Zhang, and Jiawei Han. 2020. Hierarchical topic mining via joint spherical tree and text embedding. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1908–1917. ACM.

Tomás Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2017. Advances in pre-training distributed word representations. *CoRR*, abs/1712.09405.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Maximilian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6338–6347.

John Paisley, Chong Wang, David Blei, and Michael I Jordan. 2013. A nested hdp for hierarchical topic models. *stat*, 1050:16.

Michael J Paul and Mark Dredze. 2014. Discovering health topics in social media using topic models. *PloS one*, 9(8):e103408.

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.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Dang Pham and Tuan Le. 2021. Neural topic models for hierarchical topic detection and visualization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 35–51. Springer.

Yi shi Xu, Dongsheng Wang, Bo Chen, Ruiying Lu, Zhibin Duan, and Mingyuan Zhou. Hyperminer: Topic taxonomy mining with hyperbolic embedding. In *Advances in Neural Information Processing Systems*.

Martin Stražar, Marinka Žitnik, Blaž Zupan, Jernej Ule, and Tomaž Curk. 2016. Orthogonal matrix factorization enables integrative analysis of multiple rna binding proteins. *Bioinformatics*, 32(10):1527–1535.

Yee Teh, Michael Jordan, Matthew Beal, and David Blei. 2004. Sharing clusters among related groups: Hierarchical dirichlet processes. *Advances in neural information processing systems*, 17.

Lavanya Sita Tekumalla, Priyanka Agrawal, and Indrajit Bhattacharya. 2015. Nested hierarchical dirichlet processes for multi-level non-parametric admixture modeling. *stat*, 1050:27.

Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. 2019. Poincare glove: Hyperbolic word embeddings. In *International Conference on Learning Representations*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Felipe Viegas, Sérgio D. Canuto, Christian Gomes, Washington Luiz, Thierson Rosa, Sabir Ribas, Leonardo C. da Rocha, and Marcos André Gonçalves. 2019. Cluwords: Exploiting semantic word clustering representation for enhanced topic modeling. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, pages 753–761. ACM.

Felipe Viegas, Washington Cunha, Christian Gomes, Antônio Pereira, Leonardo Rocha, and Marcos Goncalves. 2020. CluHTM - semantic hierarchical topic modeling based on CluWords. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8138–8150, Online. Association for Computational Linguistics.

Dongsheng Wang, Yi shi Xu, Miaoge Li, Zhibin Duan, Chaojie Wang, Bo Chen, and Mingyuan Zhou. Knowledge-aware bayesian deep topic model. In *Advances in Neural Information Processing Systems*.

Yiming Wang, Ximing Li, and Jihong Ouyang. 2021. Layer-assisted neural topic modeling over document networks. In *IJCAI*, pages 3148–3154.

A Additional Results

A.1 Topic Specialisation

In section 6 we report the Topic Specialisation for CluHTM and HyHTM. In this section we present the topic specialisation results in the table 5.

Dataset	**Lvl 1**	**Lvl 2**	**Lvl 3**
hLDA			
InfoVAST	0.218	0.826	0.811
Neurips	0.069	0.071	0.743
BBC	0.188	0.553	0.748
20News	0.31	0.49	0.52
Enron	0.081	0.394	0.858
Amazon	0.065	0.154	0.935
WOS46985	0.091	0.499	0.779
AGNews	0.149	0.331	0.921
TSNTM			
InfoVAST	0.08	0.19	0.28
Neurips	0.91	0.17	0.12
BBC	0.26	0.32	0.3
20News	0.31	0.49	0.52
Enron	0.18	0.29	0.38
Amazon	0.20	0.38	0.38
WOS46985	0.19	0.37	0.31
AGNews	0.22	0.50	0.67
hARTM			
InfoVAST	0.15	0.59	0.72
Neurips	0.23	0.32	0.67
BBC	0.36	0.58	0.73
20News	0.49	0.83	0.95
Enron	0.40	0.72	0.85
Amazon	0.53	0.88	0.96
WOS46985	0.42	0.81	0.96
AGNews	0.52	0.87	0.95

Table 5: Topic Specialisation for other models

B Additional Ablation Study

Hierarchical clustering with Hyperbolic Embeddings:

We replace the underlying topic model algorithm with BERTopic (Grootendorst, 2022) which uses an HDBSCAN hierarchical clustering method under the hood which does not take into account the hierarchy between words in higher-level topics and lower-level topics. Both our model and BERTopic employ hyperbolic document embeddings as A_0, followed by their respective approaches to generate a hierarchy of topics. As seen in Table 6, our model outperforms BERTopic in terms of coherence and hierarchical coherence measures. While the lower-level topics in BERTopic are related to their higher-level topics, the topic pairs (parent, child) were not unique as compared to our model.

	HyHTM	HyHTM c-TFIDF	BERTopic
Coherence	0.325	0.269	0.293
Hierarchical Coherence	0.296	0.148	0.239

Table 6: Ablation Study analyzing the effectiveness of our approach using the 20News dataset.

Investigating the Need for Post-Processing Techniques in HyHTM for Ensuring Uniqueness Across Topic Levels:

BERTopic (Grootendorst, 2022) employs a class-based TFIDF approach for topic-word representation, treating all documents in a cluster as one. Inspired by this, we examined the impact of applying a similar class-based TFIDF to topics generated by our model as an additional post-processing step. Theoretically, this should ensure unique topics at each level. However, as reported in Table 6 under **HyHTM c-TFIDF**, we found no noticeable improvement in topic coherence and hierarchy. This affirms that HyHTM inherently organizes documents into diverse and coherent themes at every level, obviating the need for additional post-processing.

C Implementation Details

C.1 Preprocessing

We remove numeric tokens, punctuations, non-ascii codes and convert the document tokens to lowercase. In addition to NLTK's stopwords, we also remove smart stopwords [5] Next we lemmatise each token using NLTK's WordNetLemmatizer. We filter the vocabulary by removing tokens whose ratio of total occurrence count to number of training

[5] Smart stopwords

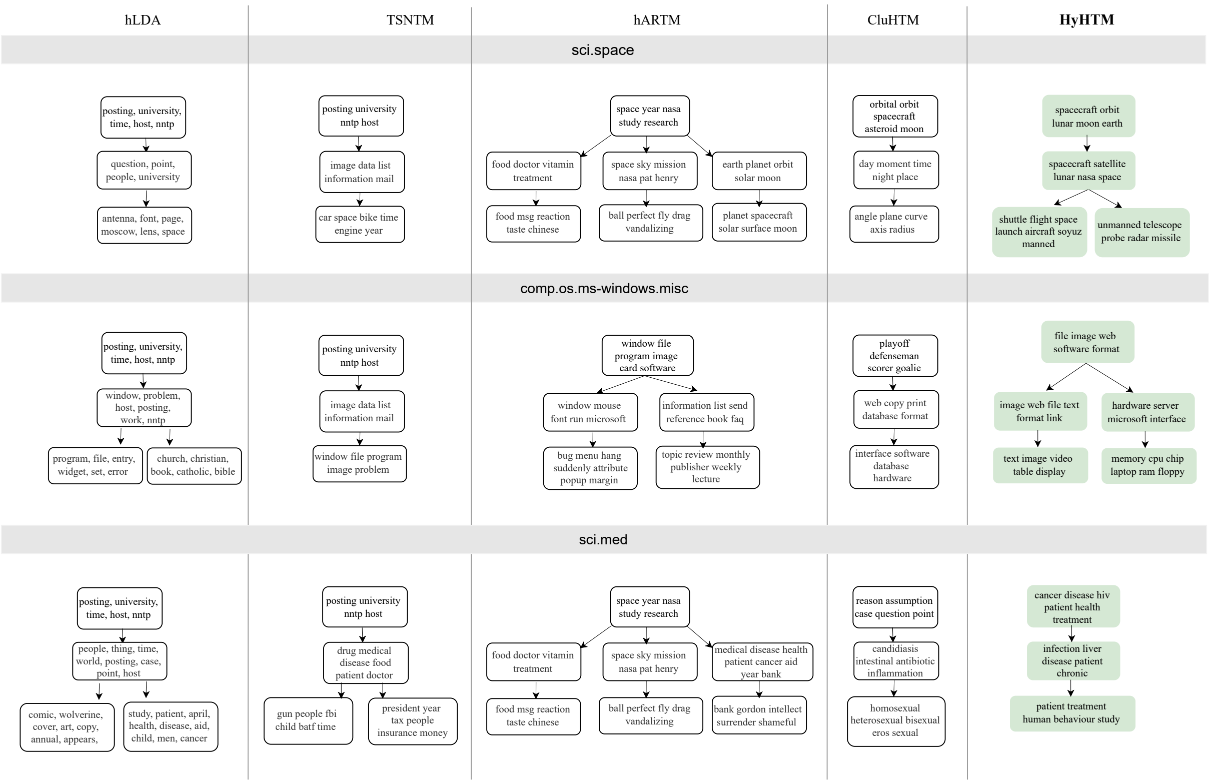

hLDA
TSNTM
hARTM
CluHTM
HyHTM
sci.space
posting, university, time, host, nntp
question, point, people, university
antenna, font, page, moscow, lens, space
posting university nntp host
image data list information mail
car space bike time engine year
space year nasa study research
food doctor vitamin treatment
space sky mission nasa pat henry
earth planet orbit solar moon
food msg reaction taste chinese
ball perfect fly drag vandalizing
planet spacecraft solar surface moon
orbital orbit spacecraft asteroid moon
day moment time night place
angle plane curve axis radius
spacecraft orbit lunar moon earth
spacecraft satellite lunar nasa space
shuttle flight space launch aircraft soyuz manned
unmanned telescope probe radar missile
comp.os.ms-windows.misc
posting, university, time, host, nntp
window, problem, host, posting, work, nntp
program, file, entry, widget, set, error
church, christian, book, catholic, bible
posting university nntp host
image data list information mail
window file program image problem
window file program image card software
window mouse font run microsoft
information list send reference book faq
bug menu hang suddenly attribute popup margin
topic review monthly publisher weekly lecture
playoff defenseman scorer goalie
web copy print database format
interface software database hardware
file image web software format
image web file text format link
hardware server microsoft interface
text image video table display
memory cpu chip laptop ram floppy
sci.med
posting, university, time, host, nntp
people, thing, time, world, posting, case, point, host
comic, wolverine, cover, art, copy, annual, appears,
study, patient, april, health, disease, aid, child, men, cancer
posting university nntp host
drug medical disease food patient doctor
gun people fbi child batf time
president year tax people insurance money
space year nasa study research
food doctor vitamin treatment
space sky mission nasa pat henry
medical disease health patient cancer aid year bank
food msg reaction taste chinese
ball perfect fly drag vandalizing
bank gordon intellect surrender shameful
reason assumption case question point
candidiasis intestinal antibiotic inflammation
homosexual heterosexual bisexual eros sexual
cancer disease hiv patient health treatment
infection liver disease patient chronic
patient treatment human behaviour study

Figure 6: Comparing topic hierarchies for 20News documents. Every topic is represented by the top most probable words of the topic.

documents in which the token appears is less than 0.8.

C.2 Computing Infrastructure

The experiments were run on a machine with NVIDIA GeForce RTX 3090 GPU and 24 GB of G6X memory. However, these experiments can also be replicated on CPU. The CUDA version used is 11.4.

C.3 HyHTM

All experiments were performed with three runs per dataset. We use the implementation provided by (Stražar et al., 2016) for NMF. With this implementation we can leverage GPUs which helps us speed the topic model. Viegas et al. (2020)'s implementation utilises the scikit-learn (Pedregosa et al., 2011) implementation of NMF. We report the difference in speed for both the approaches in Experiments 6.1.

C.3.1 Varying k_H: Neighbourhood of a word defined in the hierarchical matrix

The term k_H in equation (6) defines a neighborhood around words which helps us extract concept and sub-concept relations from hyperbolic geometry. If very large values of k_H are considered, every word would be in the neighborhood of every other word, and for very small values of k_H, even though some very similar words will be included in the neighborhood, the overall document representation will become very sparse, and many concept and sub-concept relations are discarded. We empirically tested this for k_H in the range [500, 3000], and show our findings in figure 7. We observe that when k_H is 500, the hierarchical coherence along with the other metrics, is the highest, and after that, it drops.

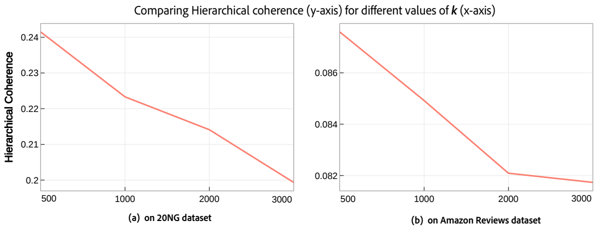
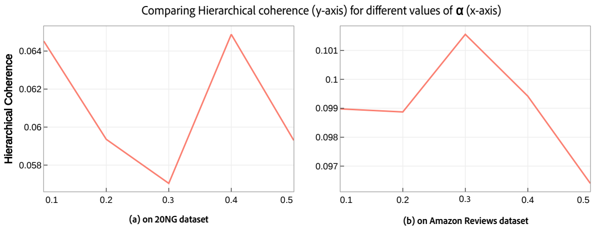


Figure 7: K_H=500 performs the best out of all the choices on Hierarchical Coherence. A similar trend is observed on other metrics as well

C.3.2 Varying α: Similarity threshold in the similarity matrix

The similarity threshold α in equation (5) is a hyperparameter that controls the pairs of words that should be considered similar and used to create the document representation. When the value is very high, only the most similar words are included in the term similarity matrix, which will result in a very sparse matrix, and defeat the purpose of adding more context about words from pretrained embeddings. If the value is very low, words which are not very similar can be picked up by the topic models as similar words. It is also important to note that while the vocabulary of terms can be controlled depending on the corpus used for topic modeling, the embeddings are pre-trained on large corpora which can result in biases from these corpora seeping into the arrangements of words in the embedding space.



(a) on 20NG dataset

(b) on Amazon Reviews dataset

Figure 8: α=0.4 performs the best out of all the choices on Hierarchical Coherence. A similar trend is observed on other metrics as well

We test our model with values of α that range from 0.1 to 0.5. In figure 8, we observe that α value 0.4 gives the maximum value of hierarchical coherence for 20ng, and α value 0.3 is the maximum for Amazon Reviews. Similarly, we fine-tuned for all other datasets and report the results in the table 7.

Dataset	α	k_h	k_S
InfoVAST	0.4	100	1000
Neurips	0.4	100	500
BBC	0.4	100	500
20News	0.1	500	500
Enron	0.4	100	500
Amazon	0.3	500	500
WOS46985	0.1	100	500
AGNews	0.1	500	500

Table 7: Best performing hyperparameters.

C.4 CluHTM

We use the implementation provided by (Viegas et al., 2020)[6] for the CLUHTM baseline. While this implementation does provide a method to learn the optimal number of topics, it is highly inefficient, taking $\mathcal{O}(n^3)$ time. The training time for this model on 20NG data was $\approx$ 32 hours, and AR was $\approx$ 22 hours. Additionally, the number of topics is different in every branch, and comparison across models becomes difficult.

C.5 hARTM

For the hARTM baseline model, we use the BigARTM[7] package, version 0.10.1. For this model, we cannot choose the number of subtopics explored for each parent, but we can control the total number of subtopics from all parents at a certain level. In our other parametric models, since each parent has n subtopics, we obtain a total of n^l topics at level l. Thus for hARTM, we indicate that the model chooses n^l topics at level l starting from $l = 1$ to a depth of $l = 3$.

C.6 hLDA

We use the following implementation[8] for hLDA.

C.7 TSNTM

We use the official implementation provided by (Isonuma et al., 2020) [9] for TSNTM.

C.8 BERTopic

We use the official implementation provided by (Grootendorst, 2022) [10] for BERTopic. We use the default parameters setup by BERTopic for HDBSCAN clustering.

D Number of topics for parametric models

For the parametric models like hARTM, CluHTM, and our model HyHTM, we use the same number of topics at every level for a fair comparison. We explain how the topic hierarchy grows when the number of topics at each node of the tree is $N = 10$.

1. At the root level (level 1), we train the model on the entire corpus of documents D and set

[6] https://github.com/feliperviegas/cluhtm

[7] BigARTM

[8] hLDA codebase

[9] TSNTM codebase

[10] BERTopic codebase

the number of topics as $N = 10$. As a result, we get 10 topics at the root level.

2. For every topic in the previous level, each parametric model organizes how documents will get distributed across topics. For CluHTM and HyHTM, a document is assigned to the topic with which it has the maximum association. Therefore, each document is assigned only 1 topic at a given level. Once the documents are categorized, we perform NMF on these documents and produce 10 topics for every parent topic.

In this way, we obtain topics at root level as 10, level 2 as $10^2 = 100$, and level 3 as $10^3 = 1000$. hARTM follows a different procedure using regularisers for categorizing documents and exploring lower-level topics. After level 1, hARTM produces flat topics in level 2 and learns the association between every lower-level topic with the higher-level topic. We assign the number of topics in level 2 as 10^2, the same as the total number of topics in level 2 for CluHTM and HyHTM, and similarity for level 3.

E Number of topics for Non-Parametric models

The number of topics for non-parametric models is listed in Table 8:

Dataset	Model	Total topics	L1 topics	L2 Topics	L3 topics
InfoVAST	hLDA	15	1	4	10
	TSNTM	12	1	5	6
Neurips	hLDA	6	1	1	4
	TSNTM	14	1	4	9
BBC	hLDA	35	1	7	27
	TSNTM	8	1	3	4
20News	hLDA	122	1	14	107
	TSNTM	20	1	7	12
Enron	hLDA	194	1	15	178
	TSNTM	9	1	3	5
Amazon	hLDA	395	1	16	378
	TSNTM	11	1	4	6
WOS	hLDA	38	1	8	29
	TSNTM	11	1	4	6
AGNews	hLDA	344	1	16	327
	TSNTM	14	1	5	8

Table 8: Number of topics for non-parametric models

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 8

☑ A2. Did you discuss any potential risks of your work?
Section 8

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract, Introduction (Section 1)

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4.1 - hyperbolic embeddings Section 5 - datasets and baseline code

☑ B1. Did you cite the creators of artifacts you used?
Section2, Section 5 Appendix C, F

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Datasets are public benchmark datasets and all code to run baselines are open source.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We have used public benchmark datasets that have been used for topic modelling.

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
We do not use such information pertaining to our datasets to analyse the quantitative performance of our models and hence leave it out for the rest of our datasets as it does not give any additional information. We use metrics that are agnostic to such characteristics and only rely on word-level statistics. Further, these are public benchmark datasets which we have provided links and citations to.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 5

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

Section 6.1

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix C, D

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix C.3.2 and C.3.3

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Appendix C.3

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix C.1

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.