Seeded Hierarchical Clustering for Expert-Crafted Taxonomies

Anish Saha[1], Amith Ananthram[1], Emily Allaway[1], Heng Ji[2], Kathleen McKeown[1]

[1]Columbia University
[2]University of Illinois Urbana-Champaign
{anish.s,amith.ananthram}@columbia.edu
{eallaway,kathy}@cs.columbia.edu, hengji@illinois.edu

Abstract

Practitioners from many disciplines (e.g., political science) use expert-crafted taxonomies to make sense of large, unlabeled corpora. In this work, we study Seeded Hierarchical Clustering (SHC): the task of automatically fitting unlabeled data to such taxonomies using only a small set of labeled examples. We propose HIERSEED, a novel weakly supervised algorithm for this task that uses only a small set of labeled seed examples. It is both data and computationally efficient. HIERSEED assigns documents to topics by weighing document density against topic hierarchical structure. It outperforms both unsupervised and supervised baselines for the SHC task on three real-world datasets.

1 Introduction

Practitioners across a diverse set of domains that include web mining, political science and social network analysis rely on machine learning techniques to understand large, unlabeled corpora (Alfonseca et al., 2012; Grimmer, 2010; Yin and Wang, 2014). In particular, they often need to fit this data to taxonomies (i.e., hierarchies) constructed by non-technical domain experts using only a few labeled examples. In this work, we formalize this task, **Seeded Hierarchical Clustering (SHC)**, and propose a novel algorithm, **HIERSEED**, for it.

Consider a researcher analyzing social media to track public feeling around a hierarchy of well-being indicators (see Figure 1). Working with such taxonomies can be challenging. Since they are hand-crafted by domain experts to explore a particular area of focus, they may be **unbalanced** (with subtopics that over-represent one aspect of their parent topic) or **incomplete** (with subtopics that are only partially enumerated). Moreover, these hierarchies may not fully explain every document in large, diverse corpora. Finally, given their domain specificity, producing many labeled examples for each topic in such taxonomies can be expensive. **SHC** incorporates these challenges as constraints: given only a user-defined topic hierarchy and a few labeled examples, the task is to assign documents from a much larger corpus to the individual topics.

While many unsupervised techniques and their hierarchical extensions automatically discover latent structure within text corpora, they are difficult to integrate with user-defined taxonomies (e.g. Blei et al. (2003); Lloyd (1982); Campello et al. (2013)). Moreover, as these methods often rely on centroids, density metrics and maximum likelihood objectives to discover dataset partitions, they may produce clusters that favor the *denser, semantically more coherent* regions of an unbalanced taxonomy at the expense of the *sparser but more diverse* regions. Although supervised hierarchical methods avoid these issues, they are usually very data intensive.

To address these challenges, we propose HIERSEED, a weakly supervised hierarchical method for fitting a large unlabeled corpus to a user-defined taxonomy. It assigns documents to topics by weighing document density against a topic's local hierarchical structure. To accommodate imbalance or incompleteness, HIERSEED constructs and uses topic representations that account for subtopic *density* (degree of semantic coherence) and *spread* (degree of semantic divergence) around each topic. As it uses only a few labeled *seed* examples to optimize its objective in a non-parametric fashion, it is both data and computationally efficient. We evaluate HIERSEED on three real-world newswire and scientific datasets and show that it outperforms state-of-the-art unsupervised and supervised baselines on this new, difficult task.

Our contributions are: (1) we **formalize the task** of **Seeded Hierarchical Clustering**, (2) we present HIERSEED, a novel algorithm that uses only a few labeled examples to efficiently fit a large corpus to a user-defined hierarchy (even if it is **unbalanced** or **incomplete**), (3) we show it **outperforms** existing methods on three real-world datasets from different

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 1595–1609
December 7-11, 2022

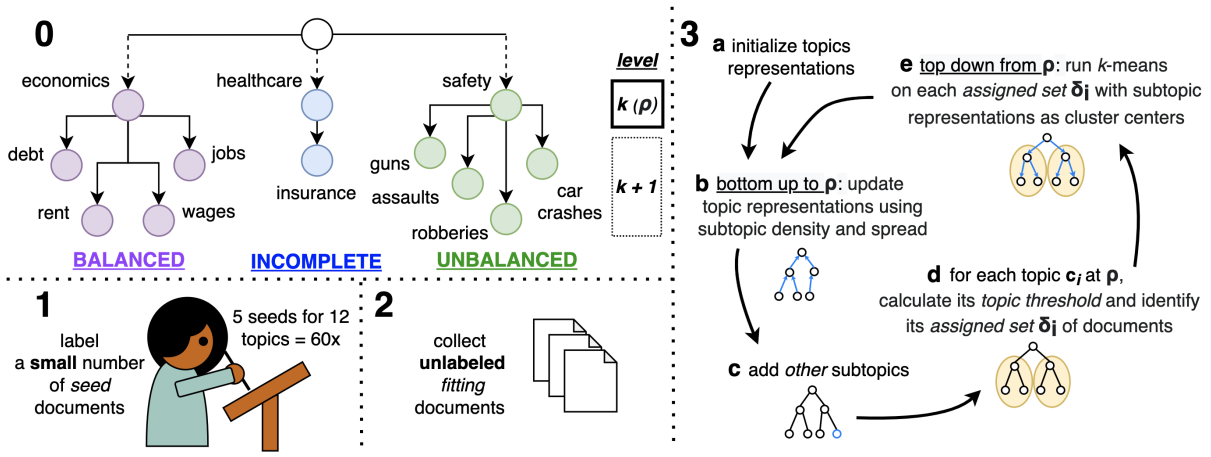


Figure 1: A researcher wants to track public well-being using a large, unlabeled social media corpus. She creates a taxonomy of relevant topics (**0**) – it does not cover every document in her dataset. Moreover, as it is hand-crafted, it is *unbalanced* and *incomplete*. She can't annotate a large number of examples. With only a few *labeled seed* examples for each topic (**1**) and her large *unlabeled fitting* set (**2**), HIERSEED efficiently identifies the documents related to every topic via an iterative discriminative algorithm that balances their density against their spread (**3**).

domains and (4) we release an **implementation of HIERSEED**[1] for the broader research community.

2 Related Work

Unsupervised methods like LDA and K-Means (Blei et al., 2003; Lloyd, 1982; MacQueen et al., 1967) are flat clustering techniques that have been successfully extended to hierarchies (Griffiths et al., 2003; Isonuma et al., 2020). While both we and Chen et al. (2005) apply K-Means iteratively, they rely on hierarchical clustering to discover the number of topics at each level. None of these methods can detect a user-defined hierarchy. There is work on taxonomy construction and expansion (Hearst, 1992; Wang and Cohen, 2007; Shen et al., 2018; Lee et al., 2022) though it cannot be used for document assignment.

Supervised hierarchical classification techniques can be categorized into flat, local and global approaches (Silla and Freitas, 2011). Flat approaches (Hayete and Bienkowska, 2005; Barbedo and Lopes, 2006) ignore the hierarchy, whereas local approaches (Koller and Sahami, 1997; Shimura et al., 2018) rely on multiple local classifiers, propagating errors down the hierarchy. Global approaches (Zhou et al., 2020; Huang et al., 2021) encode the entire hierarchy and predict all labels at once. Wang et al. (2022) generate their own text embeddings while directly embedding the hierarchy information into their encoder using contrastive learning for downstream classification. This leads to better performance and has become common for this task. Nevertheless, these approaches require a large number of labeled training examples for good performance, making them difficult to use in data-scarce scenarios.

Semi-supervised hierarchical methods require much less labeled data (Mao et al., 2012; Gallagher et al., 2017; Xiao et al., 2019). JoSH (Meng et al., 2020) uses a tree and text embedding model to jointly embed a taxonomy and a corpus in a spherical space, using category names to mine words relevant to each topic. Weakly-supervised classifiers like TaxoClass (Shen et al., 2021) and HClass (Meng et al., 2019) leverage provided keywords and documents for each topic to generate a set of pseudo documents for pretraining, then self-train on unlabeled data. Despite their strengths, these methods require more labeled data and expensive finetuning. Moreover, they work best if the labeled dataset represents all categories and documents. However, in some domains (e.g., Figure 1) knowledge of relevant categories is incomplete.

In contrast to this prior work, seeded clustering may be preferable as it first uses a small labeled seed set to bias the search space towards a desirable region and then leverages the representative latent structure of a larger unlabeled fitting set to improve the clustering (Basu et al., 2002). We propose one such technique, HIERSEED, which takes any taxonomy (unbalanced or incomplete) and a few labeled seeds and learns a discriminative hierarchical repre-

[1]Code can be found at https://github.com/anishsaha12/HierSeed

Algorithm 1: HIERSEED

Input: A corpus $\mathcal{D}$; seeds $\mathcal{S}$ and taxonomy $\mathcal{T}$ of height N; pivot level ρ.

Output: Learned topic representations for all $c_i \in \mathcal{T}$; set of relevant documents $\delta_i \subset \mathcal{D}$ for each c_i.

Initialize topics $c_i \in \mathcal{T}$ with mean of seed documents $\mathcal{S}_i$ embeddings

while *Equation 4 is minimized* **do**

  M-bottom-up, l from $N-1$ to ρ:

    update $c_i \in C^l$ with Eq. 1

  Extend taxonomy, $\forall c_i$ **Add "other"** topic to $\mathrm{ch}(c_i)$ (Eq. 2)

  E-at-level-ρ, for each $c_i \in C^\rho$:

    calc. topic threshold $\tau(c_i)$ (§3.3.2)

    identify documents δ_i (Eq. 3)

  M-top-down, l from ρ to N, $c_i \in C^l$:

    run K-means on δ_i, $k = |\mathrm{ch}(c_i)|$

    set $\mathrm{ch}(c_i)$ to K-means cluster centers

  E-top-down, l from ρ to N, $c_i \in C^l$:

    set δ_i to K-means clusters

sentation that allows assigning relevant documents to each of its component topics.

3 Methodology

We propose HIERSEED, a weakly-supervised algorithm for **Seeded Hierarchical Clustering** that uses document embeddings and their latent structure to represent topics in the same embedding space (§3.1). We initialize the representation for each topic using its seed documents (§3.2) and then update the representation in a bottom-up manner by considering both a topic's children and the density of documents around it (§3.3). Finally, we balance the hierarchy and assign documents to each topic in a top-down manner (while also updating its representation). We repeat this iteratively until convergence (see Algorithm 1, Figure 1 (3)).

3.1 Definitions

Problem Formulation Given an unlabeled corpus $\mathcal{D}$ (the *fitting set*), a hierarchy of topics $\mathcal{T}$[2] of height N and a *seed* documents set $\mathcal{S}$ for each topic, the aim of **Seeded Hierarchical Clustering** is to assign documents to their relevant topics in $\mathcal{T}$.

Let $d_i \in \mathcal{D}$ be unlabeled *fitting* documents, $c_i \in \mathcal{T}$ topics, and let $\mathcal{S}_i$ be a set of labeled *seed* documents for topic c_i. The aim is to find the set of documents $\delta_i \subset \mathcal{D}$ most relevant to each c_i. Here, a document may belong to multiple topics.

Note that we use C^l to denote all topics at level l. We denote children of a topic c_i as $c_j^{(i)} \in \mathrm{ch}(c_i)$.

Background The **Largest Empty Sphere (LES)** (Schuster, 2008) on a set of points $\mathcal{P}$, is the largest d-dimensional hypersphere containing no points from $\mathcal{P}$ but centered within its convex hull. In HIERSEED, for each topic c_i, we calculate $\mathrm{LES}(\mathrm{ch}(c_i))$, the center of the LES on c_i's subtopics (i.e., $\mathrm{ch}(c_i)$) (see Figure 2).

$\mathrm{LES}(\mathrm{ch}(c_i))$ has a particularly desirable property. Since it is as far as possible from all of its subtopics, but not too far from any particular subtopic, while also lying inside the subtopic convex hull, it helps ensure a more evenly spread surrounding document density. The main topic is adequately desensitized to any particular subtopic cluster. Recall the well-being taxonomy from Figure 1. The *safety* subtree is unbalanced – 3 of its subtopics are semantically related (*guns*, *assaults*, *robberies*). Using the centroid to represent *safety* would therefore overly favor documents related to violence at the expense of documents related to *car crashes* (an enumerated subtopic) or *workplace accidents* (an unenumerated but relevant subtopic). As $\mathrm{LES}(\mathrm{ch}(\textit{safety}))$ is informed by its subtopic spread, it is less sensitive to this imbalance.

3.2 Topic Initialization

We obtain document representations by passing each document through a word-embedding model. The representation of each topic c_i is initialized as the mean of the embeddings of all seed documents corresponding to that topic.

Let the level of a topic be $\lambda(c_i)$[3]. We choose a **pivot level** ρ, in a hierarchy, such that our algorithm computes representations and performs clustering only for topics at the pivot level or below (i.e., $\lambda(c_i) \geq \rho$). This hyperparameter may be set experimentally or through domain knowledge.

The pivot is useful because user generated hierarchies tend to become imbalanced or incomplete at lower levels. Therefore, ρ lets us choose an intermediate level such that all topics with $\lambda(c_i) < \rho$ are considered to be "complete" or fully represented by their children and can be derived without seeds. For example, in the taxonomy in Figure 1, the topics above level k are well defined. However, down

[2] We assume the hierarchy is relevant to the corpus.

[3] As is standard, λ increases as we move down the tree.

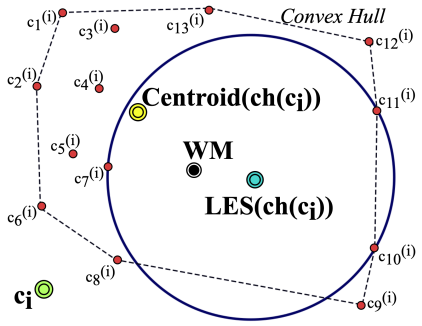


Figure 2: The topic c_i, its children $c_j^{(i)} \in \text{ch}(c_i)$, their centroid $\text{centroid}(\text{ch}(c_i))$, the center of $\text{LES}\big(\text{ch}(c_i)\big)$ and their weighted-mean WM.

the hierarchy, the topics start getting *sparse* or *unbalanced*. Thus, level k serves as a good ρ.

As a result, our proposed algorithm *only* uses seed documents for topics with levels $\lambda(c_i) \geq \rho$, further reducing the labeled data required.

3.3 Learning Topic Representations

Generally, topics lower in the hierarchy are specific and fine-grained with cohesive seeds. In contrast, top level topics are coarser with seeds that are often scattered. Thus, we need to obtain better representation at these top levels while ensuring representativeness of descendent topics.

To do so, we update each non-leaf topic representation in a bottom-up fashion as a function of: itself, its children and their "spread". Let C^l be all topics at level l. Then, for each level l from the penultimate level to ρ, for each $c_i \in C^l$, update:

$$c_i = \text{WM}\Big(c_i, \text{centroid}\big(\text{ch}(c_i)\big), \text{LES}\big(\text{ch}(c_i)\big)\Big) \tag{1}$$

Here, WM is a weighted-mean (see Figure 2), $\text{ch}(c_i)$ is the set of children of c_i, $\text{centroid}\big(\text{ch}(c_i)\big)$ is their centroid and $\text{LES}\big(\text{ch}(c_i)\big)$ is center of the *Largest Empty Sphere* formed by these children (§3.1). Since the dimension of our embedding space d is large compared to the number of child topics, computing the LES is intractable. Therefore, we propose an approximate method to estimate the center of the LES (see Appendix B).

This serves as a good updated representation of c_i. Though informed by the centroid of its subtopics, it avoids favoring topics that happen to be close to each other (i.e., denser) in an unbalanced taxonomy. It weighs finding a space that is relatively empty (LES) against a space that is relatively dense (centroid) (see Figure 2). This produces topic representations robust to hierarchy imbalances.

3.3.1 Extending Taxonomy - Other Category

A topic c_i may also be unbalanced at a particular level l if its children $c_j^{(i)} \in \text{ch}(c_i)$ are unevenly distributed around it. Alternatively, it may just be incomplete due to partial enumeration. Since topic representation updates are propagated up the taxonomy, such a topic would result in a bad representation and not only degrade the performance at that level, but propagate the imbalance upward.

One reason for this imbalance could be "incompleteness" of $\mathcal{T}$ (i.e., the set of sub-topics is not fully enumerated). Therefore, we introduce an "Other" topic $c_{\text{other}}^{(i)}$ as a subtopic of topic c_i which accounts for its missing subtopics and balances out the hierarchy. In particular, we calculate the density of the original subtopics with respect to the main topic and then define $c_{\text{other}}^{(i)}$ such that it pulls the density in the direction opposite to the centroid of the original subtopics in order to have a more even distribution of subtopics (see Figure 3).

The magnitude of $c_{\text{other}}^{(i)}$ (i.e., $\|c_{\text{other}}^{(i)}\|$) is approximated to be the magnitude of the centroid of the subtopics. Its representation is given by Equation 2 (see Appendix A for derivation). It depends on both its sibling subtopics as well as its parent topic. We extend each c_i in the taxonomy with $c_{\text{other}}^{(i)}$ in a bottom up manner from the leaf level and stop at ρ (as we define the topics above ρ to be complete).

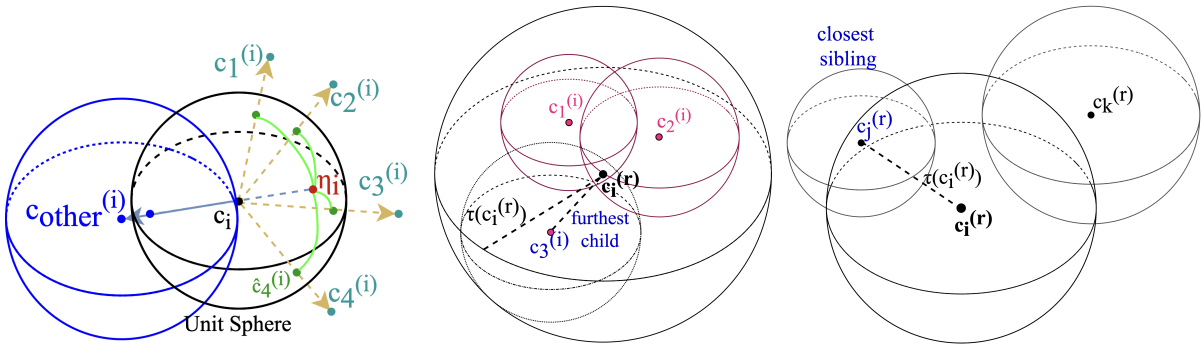
In our well-being taxonomy from Figure 1, we can see why expansion via the addition of the "Other" category is desirable. Both the *healthcare* and *safety* subtrees are only partially enumerated. Automatically expanding their subtopic sets avoids having to fully and painstakingly enumerate them.

$$c_{\text{other}}^{(i)} \approx c_i - \|c_{\text{other}}^{(i)}\| \sum_{c_j^{(i)} \in \text{ch}(c_i)} \frac{c_j^{(i)} - c_i}{\|c_j^{(i)} - c_i\|} \tag{2}$$

3.3.2 Topic Threshold and Assignment

To assign documents to topics, we initially perform distance based classification independently for each subtree at level ρ. We perform classification only at level ρ as there is a greater degree of confidence that the discovered documents actually belong to that topic since we define all topics c_i with $\lambda(c_i) < \rho$ as complete and balanced.

Let a *root* topic $c_r \in C^{\rho-1}$ be a topic at level $\rho - 1$. For each child topic $c_i^{(r)} \in \text{ch}(c_r)$ we fit a set of documents $d_k \in \delta_i^{(r)}$ belonging to it. A document belongs to the topic if it is within a certain



(a) Extending with "Other" Category. (b) Topic with children (c) Topic without children

Figure 3: (a) Extending the taxonomy by expanding the children of c_i with "Other" $c^{(i)}_{\text{other}}$. η_i is the centroid of its sub-topic unit vectors. (b, c) Topic threshold $\tau(c^{(r)}_i)$ for topic $c^{(r)}_i$ equal to (b) twice the distance of its furthest child when $\text{ch}(c^{(r)}_i) \neq \emptyset$ and (c) the distance of the nearest sibling, where $c^{(r)}_i, c^{(r)}_j, c^{(r)}_k \in \text{ch}(c_r)$, when $\text{ch}(c^{(r)}_i) = \emptyset$.

distance threshold (**topic threshold**) in the embedding space, thus partitioning the entire dataset.

$\tau(c^{(r)}_i)$ is the learned threshold distance (or a maximum radius) for each topic $c^{(r)}_i \in \text{ch}(c_r)$ when assigning documents to it. It takes into account both the density of documents $d_k \in \delta^{(r)}_i$ around $c^{(r)}_i$ and the taxonomy $\mathcal{T}$ (siblings and children). It is only defined for topics at level ρ. See Figures 3b, 3c and Appendix C for derivation.

The initial *assigned set* $\delta^{(r)}_i$, for a topic $c^{(r)}_i$ is:

$$\delta^{(r)}_i = \{d_k \mid d_k \in \mathcal{D} \wedge \|d_k - c^{(r)}_i\| \leq \tau(c^{(r)}_i)\} \tag{3}$$

If there are no documents within the topic threshold for a topic $c^{(r)}_i$, we adapt to the document density by updating the topic's threshold to be at least equal to $\alpha \geq 1$ times the distance of the nearest document from $c^{(r)}_i$, if it is within twice the original threshold.

Finally, we update the *assigned sets* $\delta^{(r)}_i$ by reperforming the assignment. We control the *degree of overlap* between the *assigned sets* using an additional hyperparameter (see Appendix C for details).

3.4 Cluster Assignment and Optimization

Given the corpus $\mathcal{D}$, the updated taxonomy $\mathcal{T}'$ (balanced, extended), the pivot level ρ and the *assigned sets*, we propose an EM-style algorithm iterating between the assignment of the document sets δ_i (**E-Step**) and recomputing the topic representations c_i (**M-Step**). We maximize the expectation (i.e., likelihood of assigning a document to a topic assuming uniform probability) by minimizing the objective:

$$\mathcal{L} = \sum_{c_i \in \mathcal{T}'} \sum_{d_k \in \delta_i} \lambda(c_i) \cdot \|d_k - c_i\|^2 \tag{4}$$

E-Step The topic thresholds are used to determine the *assigned set* for each topic at ρ (E-Step, line 6 in Algo. 1). Next, for each topic $c_i \in C^l$ at level l with *assigned set* δ_i, we solve a K-Means formulation (by Voronoi Iterations (Lloyd, 1982)) in a top-down manner from $l = \rho$ to N. The iterations are performed over the set δ_i to fit $k = |\text{ch}(c_i)|$ clusters, with initial cluster centers $c^{(i)}_k \in \text{ch}(c_i)$.

The obtained clusters correspond to the set of assigned documents $\delta^{(i)}_k$ for topic $c^{(i)}_k$ (E-top down, line 12). The process continues top-down for all sibling and successor topics until the leaves of $\mathcal{T}'$.

M-Step As the cluster centers are updated (in E-top down), we set topic representations to corresponding cluster centroids (M-top down, line 9 in Algo. 1). Now, as topic representations are a function of themselves and their children, we compute bottom-up updates of topics (parents, up to level ρ) as discussed in §3.3 (M-bottom up, line 3).

Finally, as each topic has only one parent in the taxonomy, we complete the taxonomy by directly deriving the topics above the pivot level ρ for each document set from their pivot level assignments.

Inference At inference time, we use our learned topic representations to assign documents to each topic. We use Eq. 3 to obtain *assigned sets* δ_i using the learned topic threshold $\tau(c_i)$ for each topic c_i at level ρ. Documents not within any pivot topic threshold are assigned to a *None* category. Then, each δ_i is split up among its children $c^{(i)}_j \in \text{ch}(c_i)$ by assigning each document to its closest topic $c^{(i)}_j$ to obtain the sets $\delta^{(i)}_j$. This process is repeated in a top-down manner to the leaf topics.

	WOS	NYT	RCV1
$\|\mathcal{T}\|$	141	166	103
Height of $\mathcal{T}$	2	8	4
Training	37588	29179	23149
Seed ($\|\mathcal{S}\|$ with $\|\mathcal{S}_i\| \leq 4$)	532	290	390
Fitting ($\|\mathcal{D}\|$)	37056	28889	22759
Test	9397	7292	781265

Table 1: Datasets statistics. $|\mathcal{T}|$ is the number of topics in the taxonomy. The training and test sizes are the main data splits. The seed $\mathcal{S}$ (labeled) and fitting $\mathcal{D}$ (unlabeled) sets are subsets of the training set.

Complexity Each topic's representation is updated using Eq. 1 which relies on its children and their LES, with a complexity of $\mathcal{O}(|\delta|B^3)$ where δ is the set of documents assigned to it and B is the hierarchy's maximal branching factor (see Appendix B). Each topic is then extended using Eq. 2 with complexity $\mathcal{O}(B)$, and its *topic threshold* and *assigned set* are obtained with Eq. 3 with complexity $\mathcal{O}(B + D)$ where D is the size of the unlabeled corpus. The objective Eq. 4 identifies a topic's cluster in $\mathcal{O}(D)$. We do these at most n times, once for each of our n topics, until convergence. In practice we found that convergence is achieved within 4 iterations. Overall, HIERSEED scales linearly with taxonomy size n and corpus size D.

4 Experiment Details

4.1 Datasets

We use three publicly available datasets for evaluation: RCV1-V2 (Lewis et al., 2004), NYTimes (NYT) (Sandhaus, 2008) and Web-of-Science (WOS) (Kowsari et al., 2017). RCV1-V2 and NYT are news categorization corpora while WOS includes categorization of published scientific paper abstracts. All documents in WOS belong to a single leaf topic while documents in NYT and RCV1 may belong to multiple leaf/non-leaf topics. Data statistics are shown in Table 1.

We use the benchmark train/test split for RCV1 and for NYT and WOS we randomly split the data. For each dataset, the training set is also split into the seed ($\mathcal{S}$) and fitting ($\mathcal{D}$) sets by randomly sampling a fixed number of documents $|\mathcal{S}_i|$ per topic c_i as seeds. We only keep the labels for the much smaller seed sets and discard them for the fitting sets.

4.2 Metrics

We evaluate our algorithm using B^3 (Bagga and Baldwin, 1998) and V-Measure (Rosenberg and Hirschberg, 2007). B^3 is a cluster evaluation metric that measures precision and recall of a topic distribution. V-Measure is a conditional entropy metric which measures cluster homogeneity and completeness. For both metrics, we average across all levels, weighted equally. For a fair comparison, we report the same metrics for both our method and the baselines (instead of classification F1).

4.3 Hyperparameters and Baselines

We use a pretrained RoBERTa-base (Liu et al., 2019) model to obtain a 768-dimensional embedding for each document by taking the mean across the final hidden states of all tokens. Other hyperparameters are listed in Appendix D.

We compare HIERSEED to hierarchical classification, clustering and topic modeling baselines. As a baseline, we also compare it to HIERSEED trained without the unlabeled fitting data (**Unfit-HIERSEED**). That is, Unfit-HIERSEED uses just the seed set for initial topic representations followed by lines 3-5 of Algo. 1 to update them, and line 6-8 to assign documents. We use only the labeled seed set ($\mathcal{S}$) for baselines requiring seeds or supervision and the unlabeled fitting set ($\mathcal{D}$) for unsupervised baselines. We evaluate each model 5 times and report their averages.

For weakly-supervised and unsupervised baselines we use: **hLDA** (Griffiths et al., 2003) – an unsupervised non-parametric hierarchical topic model and **TSNTM** (Isonuma et al., 2020) – an unsupervised generative neural topic model, trained on the unlabeled fitting set; **HClass** (Meng et al., 2019) – a hierarchical classification model that uses keywords from the seed set for pretraining and the unlabeled fitting set for self-training; and **JoSH** (Meng et al., 2020) – a generative hierarchical topic mining model that uses the taxonomy for supervision, trained on the unlabeled fitting set. JoSH is the only seeded hierarchical method used for comparison.

We additionally compare to a number of supervised approaches: **HDLTex** (Kowsari et al., 2017) – a hierarchical classification model trained with the labeled seed set, **HiAGM** (Zhou et al., 2020) – a hierarchical text classification model trained with the labeled seeds, **HiLAP-RL** (Huang et al., 2021) – a hierarchical classification technique trained with reinforcement learning, and **HFT** (Shimura et al., 2018) – a hierarchical CNN-based text classifier, trained on the seed set. Further details about the baselines can be found in Appendix E.

		No Fitting					Fitting + Seed				
		♦HDLTex	♦HiAGM	♦HiLAP -RL	♦HFT(M)	♦★**Unfit-HierSeed**	♠HClass	♣hLDA	♣TSNTM	♠★JoSH	♠★**HierSeed**
B^3 F1	**WOS**	0.2349	0.4044	0.1858	0.3055	0.5414	0.6420	0.1972	0.2262	0.5940	**0.7131**
	NYT	0.4840	0.4065	0.3451	0.4079	0.5040	0.5008	0.3811	0.3349	0.4692	**0.6173**
	RCV1	0.4231	0.4567	0.4279	0.5041	0.4923	0.6034	0.3873	0.3726	0.5366	**0.6546**
V-Ms	**WOS**	0.0793	0.3467	0.1383	0.2729	0.5569	0.5984	0.1984	0.1916	0.5927	**0.7661**
	NYT	0.2253	0.2264	0.1098	0.1562	0.4316	0.4461	0.1781	0.2052	0.4447	**0.5340**
	RCV1	0.1614	0.2754	0.1602	0.3422	0.3952	0.4289	0.2336	0.3026	0.3591	**0.4815**

Table 2: B^3 F1 and V-Measure (V-MS) on the WOS, NYT, RCV1 datasets. Methods are trained either using only the seed set (No Fitting) or the seed set and unlabeled fitting data (Fitting + Seed). Methods are ♦supervised classification, ♠weakly-supervised, and ♣unsupervised. ★ indicates seeded. There are up to 4 seeds per topic.

	B^3 F1			V-Measure		
	WOS	**NYT**	**RCV1**	**WOS**	**NYT**	**RCV1**
▲HDLTex	0.7310	0.6483	0.6163	0.6446	**0.6820**	0.4155
▲HiAGM	**0.7528**	0.5802	0.6363	0.6531	0.5904	0.3897
▲HiLAP-RL	0.5641	0.5116	0.6438	0.5430	0.3083	0.4037
▲HFT(M)	0.6943	**0.7130**	**0.6902**	0.7372	0.6133	**0.5472**
▼**HierSeed**	0.7131	0.6173	0.6546	**0.7661**	0.5340	0.4815

Table 3: Results of training the classification baselines with - ▲the full training set with their labels, compared to HierSeed trained using only ▼the labeled seed and unlabeled fitting sets, using 4 seeds per topic. Table 1 mentions the sizes of these sets for each dataset.

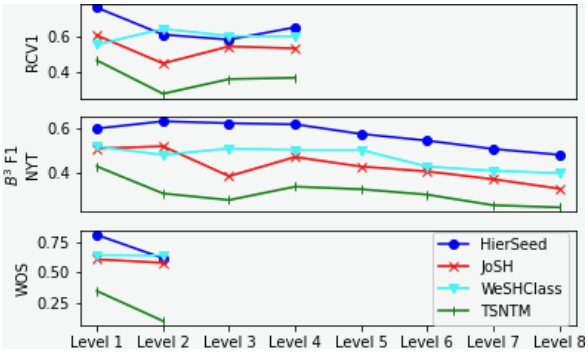


Figure 4: Performance (B^3 F1) at each level of the hierarchy for all three evaluation datasets and the best performing model from each category.

5 Results and Analysis

5.1 Main Results

Results are shown in Table 2. Our method HierSeed, trained with labeled seeds and unlabeled fitting sets, outperforms all baselines on the SHC task when restricted to the same training data. Its best score is for the WOS corpus, which we hypothesize is due to its simpler taxonomy compared to NYT and RCV1. Additionally, even without fitting on the unlabeled data our method (Unfit-HierSeed) demonstrates strong performance, outperforming most baselines. In particular, Unfit-HierSeed (which does not use fitting data) is only outperformed by the two baselines that *do* use the fitting data (HClass and JoSH), and marginally by HFT(M) on RCV1. In fact, HierSeed *with* fitting data outperforms these methods by a large margin across all corpora. This shows the effectiveness of using a small labeled seed set to fit to a taxonomy.

Although the Unfit-HierSeed outperforms most baselines, there is still a large performance drop compared to HierSeed (with fitting on unlabeled data). Since Unfit-HierSeed does not use the unlabeled data (it stops the training after the E-Step, line 6 Algo. 1, of the first iteration) it does not estimate the LES or update the *topic thresholds*. Therefore, it may inadvertently kill a branch of the hierarchy and so is limited in its ability to fit to the data. The performance drop shows the importance of LES in computing better topic representations and fitting to the provided taxonomy.

Comparing baselines, we see that the unsupervised methods (hLDA and TSNTM) perform poorly compared to the (weakly-)supervised classification methods. Although unsupervised approaches are good for discovering latent hierarchies, they aren't capable of generating topics similar to a predefined structure. Additionally, most supervised classification methods still perform poorly compared to HierSeed since these models do not make use of unlabeled data and have only a small set of seed examples for supervision.

We also examine the affect of taxonomy depth on performance. Figure 4 shows B^3 F1 at each level. Although, performance degrades at deeper levels of the hierarchy, our method is consistently better than the others at deeper levels highlighting our system's ability to learn better topic representations.

We note that although supervised classification baselines trained on the entire labeled training set outperform our method (Table 3), we achieve competitive results using a substantially smaller labeled set. The strength of our method is in the weakly-

	B^3 F1			V-Measure		
	WOS	**NYT**	**RCV1**	**WOS**	**NYT**	**RCV1**
# seed = 2	0.6701	0.6078	0.6654	0.7395	0.5021	0.5073
# seed = 4	0.7131	0.6173	0.6546	0.7661	0.5340	0.4815
# seed = 6	0.7284	0.6388	0.6870	0.7762	0.5452	0.5225
# seed = 8	**0.7288**	**0.6410**	**0.6926**	**0.7892**	**0.5538**	**0.5318**

Table 4: HIERSEED with different numbers of seeds per topic in the taxonomy. The models are trained on their respective seeds and fitted on the fitting sets.

	B^3 F1			V-Measure		
	WOS	**NYT**	**RCV1**	**WOS**	**NYT**	**RCV1**
Leaf-NoExt	0.6051	0.5609	0.6074	0.6895	0.4955	0.4164
Leaf-Ext	**0.6704**	**0.5815**	**0.6214**	**0.7263**	**0.5107**	**0.4435**
Pivot-NoExt	0.4929	0.5396	0.5836	0.5910	0.4781	0.4019
Pivot-Ext	**0.6559**	**0.5828**	**0.6138**	**0.7174**	**0.5135**	**0.4405**

Table 5: Performance of HIERSEED with (Ext) and without (NoExt) extending taxonomy with "Other" topic. Leaf: all leaf topics dropped for one penultimate parent topic. Pivot: one topic dropped at pivot level (entire subtree deleted).

supervised nature of the training procedure and it is therefore better suited to real-world data-scarce settings than fully supervised approaches.

Thus, we see the advantages of weakly-supervised approaches, and especially HIERSEED, which can both adhere to a predefined structure (i.e., a labeled taxonomy), and make good use of the much easier to obtain unlabeled fitting set.

5.2 System Analysis

We conduct analysis on the number of seeds per topic and the document representation method.

Number of Seed Examples We experiment with using different numbers of seeds for each topic in the taxonomy (see Table 4). We see a general increasing trend in the performance with an increasing number of seeds as the topics become more representative. However, the trend approaches saturation when going from 6 to 8 showcasing how little annotated data is required, to be effective.

Extending Taxonomy with Other Category Here we evaluate the effectiveness of introducing an "Other" category to balance out the "incompleteness" of the hierarchy (see Table 5). To do so, we drop a few topics from the benchmark datasets and observe the changes in the evaluation metrics and compare it with the improvement expected from introduction of the "Other" topics in the modified taxonomy. We drop both leaf and pivot-level topics, in two separate experiments, and compare the performance of HIERSEED with and without extending the taxonomy. The performance drops on deleting topics and more so when they are dropped from the pivot (as an entire subtree is deleted). However, the performance improves drastically on extending the taxonomy with our computed "Other" category in both the experiments, showing its ability to balance out incomplete hierarchies.

	B^3 F1			V-Measure		
	WOS	**NYT**	**RCV1**	**WOS**	**NYT**	**RCV1**
RoBERTa	**0.7131**	0.6173	0.6546	**0.7661**	**0.5340**	0.4815
GloVe-300d	0.7039	0.6191	**0.6692**	0.7524	0.5272	**0.5188**
fastText	0.7125	**0.6202**	0.6524	0.7574	0.5033	0.4594

Table 6: Performance of HIERSEED using different embeddings, trained using the labeled seed and unlabeled fitting sets, using 4 seeds per topic.

Embeddings for Document Representations To test HIERSEED's dependence on the nature (contextual vs. static) and quality of embeddings, we switch out the document embeddings. In Table 6, we compare the performance of HIERSEED using RoBERTa (used in all other experiments), 300-dimensional GloVe (Pennington et al., 2014), and fastText skip-gram (Joulin et al., 2016) while using 4 seeds per topic. The document embeddings are obtained by taking the mean of all tokens.

We see that HIERSEED performs equally well for each embedding, with GloVe performing better on RCV1, and fastText on NYT. However the performance differences are small, and consistently better than the baselines, showing that HIERSEED is embedding-agnostic and is able to identify a good representation for the taxonomy regardless.

5.3 Error Analysis

An analysis of HIERSEED's hierarchical assignments highlights some important shortcomings and modes of failures. First, mistakes are more likely at the pivot level than in subsequent levels. This is intuitive since taxonomies get more specific (i.e., easier to fit to) down the hierarchy and HIERSEED assigns documents top-down starting from the pivot level.

In addition, a small set of seeds for a topic may not cover all subtopics, especially if there are many semantically diverse subtopics (e.g., 'Vaccines', 'Enzymes', and 'Cancer' are diverse subtopics of 'Molecular Biology'). Furthermore, if *topics* are semantically similar (e.g., 'consumer finance' vs. 'government finance'), then the seed documents (and topic representations) may also be similar making it difficult to distinguish between the topics.

Additionally, errors come from a lack of domain specific embeddings or informative document representations. For example, corpus specific artifacts such as jargon, equations, numeric data (e.g., in WOS) and tables and figures (e.g., in NYT) can lead to uninformative document embeddings that result in incorrect topic assignment.

Finally, our method assumes an incomplete taxonomy (i.e., always adds an Other category) and therefore cannot distinguish between None and Other below the pivot level. For example, an author biography α from NYT is assigned as "Feature (level 1 - pivot level) - Books (level 2) - Other (level 3)" instead of "Feature - Books - None" (in a taxonomy consisting of just book genres). This is because, once α is assigned to the topic "Feature", it can no longer be assigned None at the following levels. However, in general we find our assumption of incompleteness is valid.

6 Conclusion and Future Work

In this paper, we formalize the task of Seeded Hierarchical Clustering: fitting a large, unlabeled corpus to a user-defined taxonomy (that may be unbalanced or incomplete) using only a small number of labeled examples. We propose a novel, discriminative weakly supervised algorithm, HIERSEED, for it which outperforms both unsupervised and (weakly) supervised state-of-the-art techniques on three real-world datasets from different domains.

In the future, we aim to jointly learn and fine-tune task specific embeddings, develop a generative variant of HIERSEED and explore non-Euclidean representation spaces.

Acknowledgements

We thank the anonymous reviewers for their comments. This work is supported in part by DARPA under agreement number FA8750-18-2-0014 and by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1644869. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements of DARPA, the U.S. Government, or the NSF. This work was also generously supported by the Research Foundation of the City of New York [CM00004929-01].

7 Limitations

While HIERSEED is non-parametric and outperforms both unsupervised and supervised baselines for the SHC task on the three real-world datasets evaluated here, it relies on a few choices and assumptions.

First, the algorithm requires selecting a pivot level below which all computations are performed. The pivot level is determined by identifying a level above which the taxonomy is well defined. For complex or incomplete taxonomies this can be hard to recognize, making the root (or a higher level) the easier choice. However, since most clustering mistakes occur at the pivot level, having a high pivot level will decrease recall at the following levels (due to error propagation) while a low pivot level will decrease precision at preceding levels as the lower levels tend to be more specific.

Another limitation is that since HIERSEED depends on external document embeddings (used to compute initial topic representations from the seeds), the clusters are sensitive to both the informativeness of the topic seeds and the richness of the embeddings. Additionally, computing representations of topics having a diverse set of subtopics is intrinsically more difficult than computing representations of topics and subtopics in a specific domain (owing to their semantic closeness). The top-down nature of topic assignment makes it crucial to start off with informative document representations to avoid error propagation. However, semantically close topics also pose a challenge as their representations become difficult to distinguish, leading to errors in topic assignment.

The form of Expectation Maximization used by HIERSEED assumes that all clusters are similarly sized and have the same variance. In practice, this may not always be the case. The use of Euclidean distance as the similarity metric, and variance as a measure of cluster scatter (as in K-Means) limits the usability in the more general Non-Euclidean cases. Additionally, HIERSEED may not be suitable for identifying clusters with non-convex shapes at each level of the hierarchy as it relies on K-Means which cannot separate non-convex clusters. However, the overall identified hierarchy may be non-convex as it is a union of multiple convex sets.

References

Enrique Alfonseca, Katja Filippova, Jean-Yves Delort, and Guillermo Garrido. 2012. Pattern learning for relation extraction with a hierarchical topic model. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 54–59.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Citeseer.

Jayme Garcia sArnal Barbedo and Amauri Lopes. 2006. Automatic genre classification of musical signals. *EURASIP Journal on Advances in Signal Processing*, 2007:1–12.

Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. 2002. Semi-supervised clustering by seeding. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, page 27–34, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer.

Bernard Chen, Phang C Tai, Robert Harrison, and Yi Pan. 2005. Novel hybrid hierarchical-k-means clustering method (hk-means) for microarray analysis. In *2005 IEEE Computational Systems Bioinformatics Conference-Workshops (CSBW'05)*, pages 105–108. IEEE.

Ryan J Gallagher, Kyle Reing, David Kale, and Greg Ver Steeg. 2017. Anchored correlation explanation: Topic modeling with minimal domain knowledge. *Transactions of the Association for Computational Linguistics*, 5:529–542.

Thomas Griffiths, Michael Jordan, Joshua Tenenbaum, and David Blei. 2003. Hierarchical topic models and the nested chinese restaurant process. *Advances in neural information processing systems*, 16.

Justin Grimmer. 2010. A bayesian hierarchical topic model for political texts: Measuring expressed agendas in senate press releases. *Political Analysis*, 18(1):1–35.

Boris Hayete and Jadwiga R Bienkowska. 2005. Gotrees: predicting go associations from protein domain composition using decision trees. In *Biocomputing 2005*, pages 127–138. World Scientific.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*.

Wei Huang, Chen Liu, Yihua Zhao, Xinyun Yang, Zhaoming Pan, Zhimin Zhang, and Guiquan Liu. 2021. Hierarchy-aware t5 with path-adaptive mask mechanism for hierarchical text classification. *arXiv preprint arXiv:2109.08585*.

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.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Daphne Koller and Mehran Sahami. 1997. Hierarchically classifying documents using very few words. Technical report, Stanford InfoLab.

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

Dongha Lee, Jiaming Shen, Seongku Kang, Susik Yoon, Jiawei Han, and Hwanjo Yu. 2022. TaxoCom: Topic taxonomy completion with hierarchical discovery of novel topic clusters. In *Proceedings of the ACM Web Conference 2022*. ACM.

David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Stuart Lloyd. 1982. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Xian-Ling Mao, Zhaoyan Ming, Tat-Seng Chua, Si Li, Hongfei Yan, and Xiaoming Li. 2012. Sshlda: a semi-supervised hierarchical topic model. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 800–809.

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

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 *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1908–1917.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 410–420.

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

Megan Schuster. 2008. The largest empty circle problem. In *Proceedings of the Class of 2008 Senior Conference, Computer Science Department, Swarthmore College*, pages 28–37.

Jiaming Shen, Wenda Qiu, Yu Meng, Jingbo Shang, Xiang Ren, and Jiawei Han. 2021. TaxoClass: Hierarchical multi-label text classification using only class names. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4239–4249, Online. Association for Computational Linguistics.

Jiaming Shen, Zeqiu Wu, Dongming Lei, Chao Zhang, Xiang Ren, Michelle T Vanni, Brian M Sadler, and Jiawei Han. 2018. Hiexpan: Task-guided taxonomy construction by hierarchical tree expansion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2180–2189.

Kazuya Shimura, Jiyi Li, and Fumiyo Fukumoto. 2018. Hft-cnn: Learning hierarchical category structure for multi-label short text categorization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 811–816.

Carlos N Silla and Alex A Freitas. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1):31–72.

Richard C Wang and William W Cohen. 2007. Language-independent set expansion of named entities using the web. In *Seventh IEEE international conference on data mining (ICDM 2007)*, pages 342–350. IEEE.

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.

Huiru Xiao, Xin Liu, and Yangqiu Song. 2019. Efficient path prediction for semi-supervised and weakly supervised hierarchical text classification. In *The World Wide Web Conference on - WWW '19*. ACM Press.

Jianhua Yin and Jianyong Wang. 2014. A dirichlet multinomial mixture model-based approach for short text clustering. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 233–242.

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.

A Derivation of Other Category

For a topic c_i and its children $\text{ch}(c_i)$, we introduce an "Other" category $c^{(i)}_{\text{other}}$ at its subtopic level to "complete" the set and denote it by $\text{ch}'(c_i)$ such that $\text{ch}'(c_i) = \text{ch}(c_i) \cup \{c^{(i)}_{\text{other}}\}$. To do so, we introduce the concept of *degree of imbalance*.

Degree of imbalance - η measures the distance of the centroid of the sub-topic representation vectors from the main topic representation, if these sub-topic were points on a unit-sphere around the main topic. A balanced hierarchy has $\eta = 0$ and an unbalanced hierarchy has η closer (never equal) to 1.

Let, $\widehat{c}^{(i)}_j$ be a unit directional vector from the topic c_i to a sub-topic $c^{(i)}_j \in \text{ch}(c_i)$. Then,

$$\widehat{c}^{(i)}_j = \frac{c^{(i)}_j - c_i}{\|c^{(i)}_j - c_i\|} \tag{5}$$

The *degree of imbalance* η_i for c_i is given by the centroid of all $\widehat{c}^{(i)}_j$ as:

$$\eta_i = \frac{1}{|\text{ch}(c_i)|} \sum_{c^{(i)}_j \in \text{ch}(c_i)} \widehat{c}^{(i)}_j \tag{6}$$

We want to decrease the *degree of imbalance* of the augmented sub-topic set $\text{ch}'(c_i)$ such that,

$$\eta'_i \approx 0$$

$$\implies \frac{1}{|\text{ch}'(c_i)|} \sum_{c^{(i)}_j \in \text{ch}'(c_i)} \widehat{c}^{(i)}_j$$

$$= \frac{1}{|\text{ch}(c_i)| + 1} \sum_{c^{(i)}_j \in \text{ch}(c_i)} \widehat{c}^{(i)}_j + \widehat{c}^{(i)}_{\text{other}}$$

$$\approx 0$$

Therefore,

$$\widehat{c}^{(i)}_{\text{other}} \approx -|\text{ch}(c_i)| \cdot \eta_i \tag{7}$$

Solving for $c^{(i)}_{\text{other}}$ using Equations 5 and 7:

$$c^{(i)}_{\text{other}} \approx -|\text{ch}(c_i)| \cdot \|c^{(i)}_{\text{other}} - c_i\| \cdot \eta_i + C_i$$

We can set the magnitude of the "Other" vector to approximately be:

$$\|c^{(i)}_{\text{other}} - c_i\| \approx \left\| \frac{1}{|\text{ch}(c_i)|} \sum_{c^{(i)}_j \in \text{ch}(c_i)} (c^{(i)}_j - c_i) \right\|$$

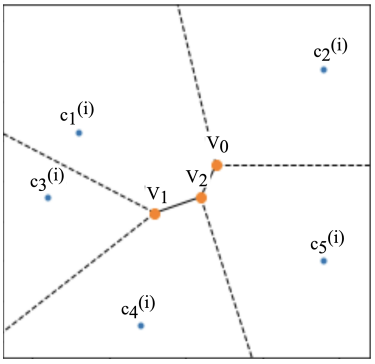


Figure 5: Voronoi diagram of a set of points $\mathcal{P} = \text{ch}(c_i)$. The Voronoi vertices are shown with V_i in the diagram. There are three such vertices here.

i.e. distance of the centroid of the sub-topics $c^{(i)}_j \in \text{ch}(c_i)$ from the topic c_i.

Therefore, on substituting the values we get,

$$c^{(i)}_{\text{other}} \approx -\eta_i \cdot \left\| \sum_{c^{(i)}_j \in \text{ch}(c_i)} (c^{(i)}_j - C_i) \right\| + c_i \tag{8}$$

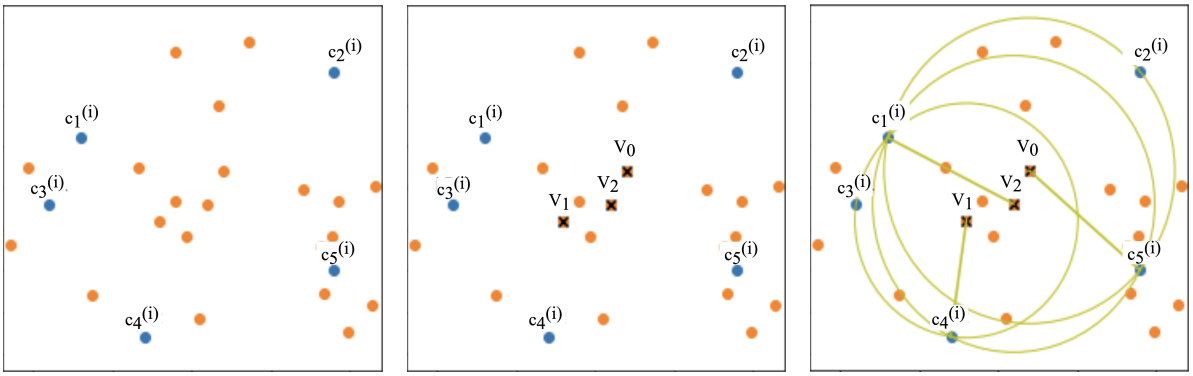
This gives us the "Other" category to augment the sub-topics set $\text{ch}(c_i)$ (Figure 3a shows the configurations in embedding space). We apply this in a bottom up manner starting from level N to ρ (pivot level) as we assume that the topics above it are complete. From the equation we see that the representation of the "Other" category depends on representations of sibling sub-topics as well as the parent topic.

To compute $c^{(k)}_{\text{other}}$ for $\text{ch}(c_k)$ such that $\lambda(c_k) = \rho - 1$ (i.e. finding the "Other" category for topics at pivot level), we do not readily have c_k. Here, we first set $c_k = \text{LES}\big(\text{ch}(c_k)\big)$, and then compute $c^{(k)}_{\text{other}}$.

B Approximating LES

The *LES* for a set of points is found by constructing the Voronoi diagram (Figure 5) which divides a space such that all points within a region are closest to a point $p \in \mathcal{P}$, than to any other point $p' \in \mathcal{P}$. The center of a *LES* is always either a Voronoi vertex or is the point of intersection of a Voronoi edge and the convex hull (Schuster, 2008).

In our case, the set of points $\mathcal{P} = \text{ch}(c_i)$ for each topic $c_i \in \mathbb{R}^d$. We know for most practical purposes $d \gg |\mathcal{P}|$ and this makes the problem intractable with no solution. Thus, we propose an approximate form of Voronoi decomposition.



(a) Sub-topics and data (b) Approx Voronoi vertices (c) Delaunay triangle Spheres

Figure 6: Approximating set of Voronoi vertices $\widehat{V}_i$ from set of sub-topic points $\mathcal{P} = \text{ch}(c_i)$ and data points $d_k \in \delta_i$ for a topic c_i. (a) data points colored orange and $\mathcal{P}$ colored blue. (b) Approx Voronoi vertices $v_k \in \widehat{V}_i$ labeled with black $\times$. $\widehat{V}_i \subset \delta_i$. (c) Circumcircles (spheres) drawn with v_k as center and $p \in \mathcal{P}$ on circumference.

Ideally, the sub-topics $\mathcal{P}$ and the assigned subset of documents δ_i for topic c_i, would be distributed such that a distance based formulation such as K-Means would converge with centers equal to $\mathcal{P}$, and the decision boundaries would construct the Voronoi diagram. We know that the documents $d_k \in \delta_i$ represent a subset of the set of infinite spatial points Φ_i around c_i, i.e., $\delta_i \subset \Phi_i$. Also, the set of Voronoi vertices $v_k \in V_i$ for topic c_i satisfies $V_i \subset \Phi_i$, and bounded by δ_i.

The duality of Voronoi diagrams and Delaunay triangulation states that Voronoi vertices are the circumcenters of Delaunay triangles, where the vertices of the triangles are from $\mathcal{P}$. That is, for every $v_k \in V_i$ there are three points $p_1, p_2, p_3 \in \mathcal{P}$:

$$\|p_1 - v_k\| = \|p_2 - v_k\| = \|p_3 - v_k\|$$

Since $V_i \subset \Phi_i$ one can iterate through all points in Φ_i to find all v_k that satisfy this equality. However, since Φ_i is infinite, we approximate V_i from δ_i instead (see Figure 6).

The approximate set of Voronoi vertices $\widehat{V}_i$ such that, $\widehat{V}_i \subset \delta_i$, and for all combination of three points $p_1, p_2, p_3 \in \mathcal{P}$ is given by:

$$\widehat{V}_i = \{d_{k'} \mid d_{k'} \in \delta_i \wedge \|p_1 - d_{k'}\| \approx \|p_2 - d_{k'}\| \approx \|p_3 - d_{k'}\|\} \quad (9)$$

Since there are $\mathcal{O}(|\mathcal{P}|^3)$ such combinations and we iterate through δ_i, the time complexity of this approximate algorithm is $\mathcal{O}(|\mathcal{P}|^3|\delta_i|)$ for each topic.

This approximation is good if the set δ_i is dense. Additionally, the error threshold for the equality in Equation 9 may be determined based on the statistics (mean and standard deviation) of all errors.

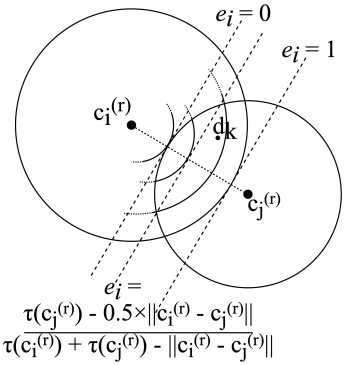


Figure 7: Three eccentricity settings e_i for the topic C_i w.r.t C_j to resolve *assigned set* overlap.

Finally, the $\text{LES}\big(\text{ch}(c_i)\big)$ is the center of the sphere with the largest radius (Figure 6c).

C Topic Threshold and Assigned Set Overlap

Topic Threshold - For each $c_i^{(r)} \in \text{ch}(c_r)$, if $\text{ch}(c_i^{(r)}) \neq \emptyset$, the topic threshold $\tau(c_i^{(r)})$ is given by Equation 10 (twice the distance of the furthest child), else, by Equation 11 (distance of the nearest sibling).

$$\tau(c_i^{(r)}) = 2 \times \max_{c_j^{(i)} \in \text{ch}(c_i^{(r)})} \|c_i^{(r)} - c_j^{(i)}\| \quad (10)$$

$$\tau(c_i^{(r)}) = \min_{c_j^{(r)} \in \text{ch}(c_r) - \{c_i^{(r)}\}} \|c_i^{(r)} - c_j^{(r)}\| \quad (11)$$

Eccentricity - there may be overlap between the *assigned sets* δ_i for each topic c_i. Thus, to control the degree of overlap between these sets, we introduce a parameter called *eccentricity* e_i, for

each of these topics c_i, such that $e_i \in [0, 1]$ (see Figure 7).

For a topic $c_i^{(r)} \in \text{ch}(c_r)$ (for a topic $c_r \in C^{\rho-1}$) and each sibling topic $c_j^{(r)} \in \text{ch}(c_r) - \{c_i^{(r)}\}$ such that $\delta_i \cap \delta_j \neq \emptyset$, consider a document $d_k \in \delta_i \cap \delta_j$. We keep d_k in δ_i if:

$$\begin{aligned} \|d_k - c_i^{(r)}\| \leq \|c_i^{(r)} - c_j^{(r)}\| - \tau(c_j^{(r)}) + \\ e_i\Big(\tau(c_i^{(r)}) + \tau(c_j^{(r)}) \\ - \|c_i^{(r)} - c_j^{(r)}\|\Big) \end{aligned} \tag{12}$$

The most common setting for e_i is:

$$e_i = \frac{\tau(c_j^{(r)}) - 0.5 \times \|c_i^{(r)} - c_j^{(r)}\|}{\tau(c_i^{(r)}) + \tau(c_j^{(r)}) - \|c_i^{(r)} - c_j^{(r)}\|}$$

which assigns d_k to $c_i^{(r)}$ if it is nearer to it, compared to $c_j^{(r)}$. This is done for all topics $c_i^{(r)} \in \text{ch}(c_r)$ to obtain the final *assigned sets* δ_i.

D Hyperparameter Settings

The topic representations (§3.3) are computed by choosing a pivot level $\rho = 2$ for all three datasets.

The weights for the weighted-mean (Equation 1) are set to 1 for the centroid term $\text{centroid}\big(\text{ch}(c_i)\big)$, 5 for the main topic c_i, and the weight of $\text{LES}(\text{ch}(c_i))$ is set to 4 for WOS, RCV1, and to 1 for the NYT dataset.

The value of α is set to 1.1 while recomputing the topic thresholds in case originally discovered *assigned sets* are empty.

For each topic pair $c_i, c_j \in \mathcal{T}$, we set the eccentricity to be:

$$e_i = \frac{\tau(c_j) - 0.5 \times \|c_i - c_j\|}{\tau(c_i) + \tau(c_j) - \|c_i - c_j\|}$$

(in Equation 12).

Finally, we allow the algorithm to extend the taxonomy at all levels, with the "Other" category, for all three datasets.

We also report our performance without fitting on the unlabeled fitting set, utilizing just the seed set for training. In Table 2, we set $c = 4$ for all three datasets, and we alleviate the randomness by repeating the seed sampling process 5 times and reporting the average metrics.

E Baselines

We compare our with multiple baselines spanning unsupervised/seed-guided hierarchical topic models, unsupervised/seed-guided text embedding models, and weakly-supervised/supervised hierarchical text classification models.

- hLDA (Griffiths et al., 2003): a non-parametric hierarchical topic model based on the nested Chinese restaurant process with collapsed Gibbs sampling, which assumes documents are generated from a word distribution of a path of topics. Since it is unsupervised we use the training set (seed+fitting set) without any labels, and obtain the dynamic topic clusters.
- TSNTM (Isonuma et al., 2020): a generative neural topic model which uses VAE inference to detect topic hierarchies in documents. Being unsupervised, we treat it as a clustering method which decides the hierarchical clusters dynamically, by fitting on the unlabeled training set.
- JoSH (Meng et al., 2020): a weakly-supervised generative hierarchical topic mining model which uses a joint tree and text embedding method to simultaneously model the category tree structure and the corpus generative process in the spherical space. We use this as a text classifier, trained on the unlabeled training set using the topic taxonomy as the supervision.
- WeSHClass (Meng et al., 2019): a weakly-supervised hierarchical classification model which leverages the provided keywords of each topic to generate a set of pseudo documents for pretraining and then self-trains on unlabeled data, using Word2Vec as embeddings. We use the keywords from the seed set for pretraining and the unlabeled fitting set for self-training.
- HDLTex (Kowsari et al., 2017): a supervised method that combines multiple deep learning approaches in a top-down manner to produce hierarchical classification, by creating specialized architectures for each level of the hierarchy. Of the multiple variants presented, we use the RNN-RNN combination and train the model with the labeled seed set.
- HiAGM (Zhou et al., 2020): an end-to-end hierarchical structure-aware global model that learns hierarchy-aware label and structure embeddings, formulated as a directed graph, which is then fused with text features to produce hierarchical text classification. We use the HiAGM-TP model

and the GCN structure encoder, and train with the labeled seeds.

- HiLAP-RL (Huang et al., 2021): a top-down reinforcement learning based approach to hierarchical classification, where modeled as a Markov decision process and learns a label assignment policy. We use HiLAP with bow-CNN as the policy model.
- HFT (Shimura et al., 2018): a hierarchical CNN fine-tuning based approach for text classification where the model learns a classifier for the upper class labels, and uses transfer learning for the lower classes, thereby directly utilizing the parental/children dependency between adjacent levels. We train the HFT-CNN model using the recommended scoring function (MSF), on the seed set.