Document Hashing with Multi-Grained Prototype-Induced Hierarchical Generative Model

Qian Zhang, Qinliang Su[*], Jiayang Chen, Zhenpeng Song
School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China,
`{zhangq637,chenjy265,songzhp3}@mail2.sysu.edu.cn`,
`suqliang@mail.sysu.edu.cn`

Abstract

Document hashing plays a crucial role in large-scale information retrieval. However, existing unsupervised document hashing methods merely consider flat semantics of documents, resulting in the inability of preserving hierarchical semantics in hash codes. In this paper, we propose a hierarchical generative model that can model and leverage the hierarchical structure of semantics. Specifically, we introduce hierarchical prototypes into the model to construct a hierarchical prior distribution, which is integrated into the variational auto-encoder (VAE) framework, enabling the model to produce hash codes preserving rough hierarchical semantics. To further promote the preservation of hierarchical structure, we force the hash code to preserve as much semantic information as possible via contrastive learning, which exploits the hierarchical pseudo labels produced during VAE training. Extensive experiments on three benchmarks outperform all baseline methods, demonstrating the superiority of our proposed model on both hierarchical datasets and flat datasets.

1 Introduction

Similarity search aims at retrieving documents of high similarity with the query input from a huge database, and has been found useful in a large number of applications like plagiarism analysis (Stein et al., 2007), collaborative filtering (Koren, 2008) etc. Semantic hashing (Salakhutdinov and Hinton, 2009) represents the documents by compact binary codes, thus is able to evaluate the similarity between two documents at the low-cost hamming space, consequently improving the retrieval speed and memory footprint compared with continuous real-valued features. One of the most widely-used approaches for unsupervised hashing is to model the documents through a deep generative model (Kingma and Welling, 2013;Chaidaroon and Fang, 2017;Shen et al., 2018), in which the semantic information of documents are encoded into the hash codes through the reconstruction of documents. After that, considerable researches have been devoted to improve the quality of generated hash codes by incorporating more semantic knowledge into the generative models like the knowledge of inherent cluster structure in data (Dong et al., 2019; Ye et al., 2020), neighborhood information among documents (Hansen et al., 2020; Ou et al., 2021a; He et al., 2023) etc. Thanks to the exploitation of these semantic knowledge, the hash codes produced by these models are shown to retain more semantic similarity information of documents.

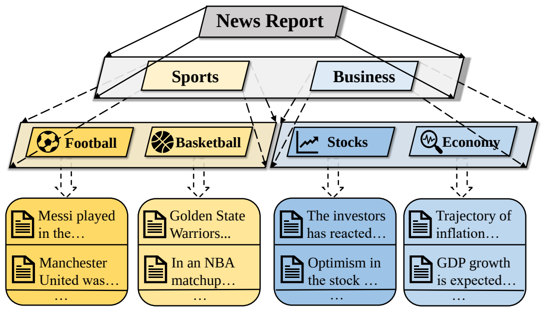


Figure 1: Hierarchical structure of categories generally occurs in datasets.

However, existing unsupervised methods (Chaidaroon and Fang, 2017;Shen et al., 2018;Ou et al., 2021a;Ou et al., 2021b;He et al., 2023) concentrate on preserving flat semantic structure during their modeling process, but rarely consider and leverage the *hierarchical* semantic structure that exists ubiquitously in real world. For instance, a group of documents related to 'sports' and 'business' can be further subdivided into finer-grained categories like 'basketball', 'football', 'stocks', 'economy', etc, or be conversely summarized into a coarser-grained category like 'news' (refer to Figure 1 for intuitive illustration). An ideal retrieval result is expected to be able to reflect the inherent hierarchical semantic

[*] Corresponding author

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 321–333
November 12-16, 2024 ©2024 Association for Computational Linguistics

structure. For example, given a query document from 'basketball', the most favorable retrieved documents should also belong to 'basketball', followed by documents from 'football', because documents from the same coarse-grained category 'sports' are more acceptable than those from different coarse-grained category, *e.g.*, 'business'. Obviously, by explicitly taking into account the *hierarchical* semantic structure in the hash model, the generated hash codes can not only reflect the desired hierarchical semantic structure, but also yield more accurate retrieval results. Yet how to effectively model the hierarchical structure of semantics for unsupervised document hashing is unexplored before. Thus, the problems confronting us can be concluded as: (1) How to model the hierarchical structure of semantics in unsupervised scenario? (2) How to skillfully integrate hierarchical structure into the framework of the generative hashing model?

To address the aforementioned problems, in this paper, we propose HierHash: a multi-grained prototype-induced **Hier**archical generative **Hash**ing model that can explicitly model and leverage the underlying hierarchical structure of semantics in documents. Specifically, coarse-grained and fine-grained prototypes are first introduced and then leveraged to construct a hierarchical prior distribution, which is later integrated into the variational auto-encoder (VAE) for deep hierarchical modeling of documents. With the coarse-/fine-grained prototypes automatically learned from the training documents, the generative model is able to produce hash codes that roughly reflect the underlying hierarchical structure of semantics. To further promote the semantic hierarchies in the learned hash codes, in addition to the training objective on the generative model, we also force the hash codes output from the model to preserve as much semantic information as possible via contrastive learning, with the help of the hierarchical pseudo labels discovered by the generative model. Finally, with the pseudo labels becoming increasingly accurate, a hierarchical self-labeling module is further introduced, which further improves the hash code quality through strong supervision from high-confidence pseudo labels. Extensive experiments of HierHash on three public benchmarks outperform all baseline methods, demonstrating its superiority on both hierarchical datasets and flat datasets. The evaluation of the coarse-grained retrieval on two hierarchical datasets also demonstrates the effectiveness on hierarchical retrieval of our model.

2 Related work

Unsupervised Document Hashing Deep generative models (Rezende et al., 2014) have attracted attention in the realm semantic document hashing, where an encoder-decoder architecture was established to encourage binary codes to retain semantic information by reconstructing original data. VDSH (Chaidaroon and Fang, 2017) proposed to learn continuous representations under variational autoencoder (VAE) framework (Kingma and Welling, 2013) with an assumption of Gaussian prior and then cast it into binary codes; NASH (Shen et al., 2018) replaced Gaussian prior with Bernoulli prior to construct end-to-end generative hashing framework; BMSH (Dong et al., 2019) employed a mixture prior; Corrsh (Zheng et al., 2020) introduced the distribution of Boltzmann machine to the generative model and WISH (Ye et al., 2020) followed NASH (Shen et al., 2018) and introduced a set of auxiliary implicit topic vectors to address the information loss in few-bits hashing. What's more, PairRec (Hansen et al., 2020) and SNUH (Ou et al., 2021a) further leveraged neighborhood information in generative model. Recently, contrastive methods were also proven effective in hashing task (Qiu et al., 2021; Qiu et al., 2022; Ou et al., 2021b). However, these approaches undergone a similar problem to ignore potential hierarchical structure of data, which widely exists in real-world datasets.

Hierarchical Hashing Some works in hashing task (Wang et al., 2017; Chen et al., 2018; Sun et al., 2019) paid attention to the hierarchical categories of datasets. For instance, IHDH (Guo et al., 2023) proposed a document hashing model to make use of neighboring information and the hierarchical structure to learn hierarchical hashing codes. But they are all supervised methods which requires strong supervision from delicately-labelled hierarchical datasets, making them ungeneralizable.

Hierarchical Structure In the realm of unsupervised image representation learning, many works (Li et al., 2020; Guo et al., 2022; Xu et al., 2022) proposed to build hierarchical prototypes to guide the learning of representations. Other researches (Yang et al., 2020; Bukchin et al., 2021; Ni et al., 2021) proposed to learn representations for downstream few-shot classification task with coarse-grained labels of data. The aforementioned works demonstrated the feasibility and significance of learning representations with multiple semantic hierarchy.

3 The Proposed Method

In this section, a deep generative model is first developed to model the hierarchical semantic structure of documents, which can explicitly encourage the formation of hierarchical structure in document representations. Then, a contrastive learning based method is further developed to strengthen the hierarchical structure by forcing the representations to better reflect the semantic information of documents, since the semantics of documents is thought to be inherently hierarchical.

3.1 Hierarchical Generative Model

To begin with, we propose a hierarchical generative model, with the joint distribution $p(x, z, s, y)$ defined as:

$$p(x, z, s, y) = p_\theta(x|z)p(z|s)p(s|y)p(y), \quad (1)$$

where x denotes the raw feature of the observed document, which is the [CLS] embedding from BERT (Devlin et al., 2019) in this paper; y denotes the coarse-grained category of document x with $p(y = m) = \frac{1}{M}$ for $m = 1, 2, \cdots, M$; random variable $s \in \{1, \cdots, K\}$ denotes the fine-grained category of document x, with K indicating the total number of fine-grained categories; and z denotes the latent representation.

The Naive Approach To model a hierarchical relationship between coarse-grained and fine-grained categories, a naive way is to designate a coarse-grained category for every fine-grained category. Obviously, this amounts to partition the fine-grained category set $\{1, \cdots, K\}$ into M disjoint subsets $\mathbb{S}_1, \cdots, \mathbb{S}_M$ with $\mathbb{S}_1 \cup \mathbb{S}_2 \cdots \cup \mathbb{S}_M = \{1, \cdots, K\}$ and $\mathbb{S}_i \cap \mathbb{S}_j = \varnothing$ for any $i \neq j$. Under such a modeling scheme, we can set the conditional pdf $p(s|y)$ as:

$$p(s|y) = \begin{cases} \frac{1}{|\mathbb{S}_y|}, & s \in \mathbb{S}_y \\ 0, & \text{else} \end{cases} . \quad (2)$$

However, since the ground-truth hierarchical structure is unknown in unsupervised scenarios, the partition of sets $\{\mathbb{S}_m\}_{m=1}^M$ that determines the distribution $p(s|y)$ is unknown and need to be learned from the training data. Theoretically, the partition problem can be solved by exhaustively searching all candidate partitions and selecting the one that best fits the dataset. However, the number of candidate partitions is the Stirling number of the second kind $S(M, K)$ (Stanley, 2011) in combinatorial mathematics, which is notoriously large as M grows, making the exhaustive search infeasible. What's worse, even if the set partition $\{\mathbb{S}_m\}_{m=1}^M$ can be learned, it merely reflects the hierarchical relationship of categories, but what we really want is to have the hierarchical structure reflected on the latent representations z.

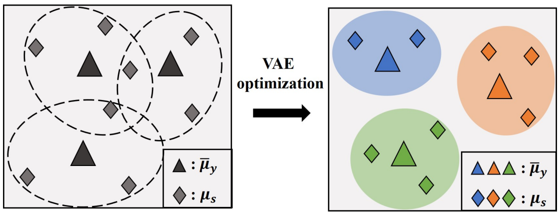


Figure 2: With the optimization, the prototypes can demonstrate hierarchical structure in latent space.

Hierarchical-Prototype-Induced Approach To overcome the limitations of naive hierarchical modeling scheme above, we propose to introduce hierarchical prototypes to model the hierarchical structure. Specifically, by denoting $\{\bar{\mu}_y\}_{y=1}^M$ and $\{\mu_s\}_{s=1}^K$ as the prototype representations of coarse-grained category y and fine-grained category s in latent space, we define the joint pdf $p(s, y)$ as $p(s, y) = p(s|y)p(y)$ with $p(s|y)$ as:

$$p(s|y) = \frac{e^{-\|\mu_s - \bar{\mu}_y\|_2^2}}{\sum_{s'=1}^K e^{-\|\mu_{s'} - \bar{\mu}_y\|_2^2}}, \quad (3)$$

where $\|\cdot\|_2$ is the L2-norm operation; the prototypes $\{\mu_s\}_{s=1}^K$ and $\{\bar{\mu}_y\}_{y=1}^M$ are learnable parameters, and can also be trained together with the generative model. As illustrated in Figure 2, by setting the coarse-grained and fine-grained prototypes $\{\bar{\mu}_y\}_{y=1}^M$ and $\{\mu_s\}_{s=1}^K$ to appropriate vectors, the conditional distribution $p(s|y)$ can effectively model the hierarchical structure between s and y according to the distance between between the fine-grained prototypes $\{\mu_s\}_{s=1}^K$ and the coarse-grained prototypes $\{\bar{\mu}_y\}_{y=1}^M$. Roughly, a fine-grained prototype μ_s is more likely to be assigned to a coarse-grained prototype $\bar{\mu}_y$ close to it. Or equivalently, a coarse-grained prototype $\bar{\mu}_y$ tends to absorb the fine-grained prototypes close to it as its sub-categories. Therefore, by modeling

$p(s|y)$ as in (3), the hierarchical structure on the categories and latent representations can be effectively modelled simultaneously.

Given the fine-grained category s, we then define the conditional distributions $p(z|s)$ and $p(x|z)$ as

$$p(z|s) = \mathcal{N}\left(z; \mu_s, \sigma_s^2 I\right) \quad (4)$$

$$p_\theta(x|z) = \mathcal{N}(x; \mu_x, \sigma_x^2 I), \quad (5)$$

where the fine-grained prototype μ_s is directly used as mean value of the Gaussian distribution $p(z|s)$; σ_s^2 is the variance; and $[\mu_x, \sigma_x]$ in distribution $p_\theta(x|z)$ are outputs of a neural network $f_\theta(z)$:

$$[\mu_x, \sigma_x] = f_\theta(z). \quad (6)$$

Since the latent variable z is drawn from the distribution centered by μ_s, which is close to $\bar{\mu}_y$ as is defined in Equation (3), z is close to both μ_s and $\bar{\mu}_y$, equipping itself with the ability to preserve the semantics of both coarse-grained and fine-grained category. The document x generated based on z is therefore correlated with the hierarchical semantics. To conclude, with $p(s, y)$ introducing hierarchical categories into the model and $p(z|s), p(x|z)$ associating the document with the categories, our proposed generative model is able to unify the hierarchical structure of semantics and the generative model.

3.2 Training

To train the hierarchical generative model, we first maximize the evidence lower bound (ELBO) of $\log p(x_i)$, which is formulated as follows:

$$\mathcal{L}_{\text{ELBO}} = \sum_{i=1}^{N} \mathbb{E}_{q_\phi(y_i,z_i,s_i|x_i)}\left[\log \frac{p(x_i, y_i, s_i, z_i)}{q_\phi(y_i, z_i, s_i|x_i)}\right], \quad (7)$$

where $q_\phi(y_i, z_i, s_i|x_i)$ is the joint variational posterior. $\mathcal{L}_{\text{ELBO}}$ can be divided into four different parts: $\mathbb{E}_{q_\phi(z_i|x_i)}[\log p_\theta(x_i|z_i)]$, $\mathbb{KL}(q_\phi(z_i|x_i)\|p(z_i|s_i))$, $\mathbb{KL}(q(s_i|z_i)\|p(s_i|y_i))$ and $\mathbb{KL}(q(y_i|s_i, z_i)\|p(y_i))$, the detailed derivation of which is in appendix A.1.

To calculate $\mathcal{L}_{\text{ELBO}}$, the variational posterior needs to be defined, which is factorized as:

$$q_\phi(y_i,z_i,s_i|x_i) = q_\phi(z_i|x_i)q(s_i|z_i)q(y_i|s_i,z_i). \quad (8)$$

$q_\phi(z_i|x_i)$ is defined as a multivariate Gaussian distribution $\mathcal{N}(z_i; \tilde{\mu}_i, \tilde{\sigma}_i^2 I)$ in which $\tilde{\mu}_i$ and $\tilde{\sigma}_i$ are specified by a neural network $g_\phi(\cdot)$:

$$[\tilde{\mu_i}; \tilde{\sigma_i}] = [g_{\phi_1}(x_i); g_{\phi_2}(x_i)]. \quad (9)$$

Thus, the latent representation z_i for document x_i can be reparameterized as

$$z_i = \tilde{\mu}_i + \xi_i \cdot \tilde{\sigma}_i, \quad (10)$$

where ξ_i is drwan from Gaussian distribution: $\xi_i \sim \mathcal{N}(0, I)$. With $q_\phi(z_i|x_i)$ derived, the analytical forms of the first two terms of $\mathcal{L}_{\text{ELBO}}$ can be derived. Since they are similar to the VAE optimization process in classical generative-based hashing (Chaidaroon and Fang, 2017), we introduce the details in appendix A.2 and A.3.

As for $q(s_i|z_i)$, to make sure it can reflect the probability of z_i being assigned to the fine-grained category s_i, we propose to relate it to the semantic similarity between z_i and μ_s:

$$q(s = j|z_i) = \frac{e^{-\|z_i-\mu_j\|_2^2/\tau_s}}{\sum_{j'=1}^{K} e^{-\|z_i-\mu_{j'}\|_2^2/\tau_s}}, \quad (11)$$

where τ_s denotes the temperature. $q(y_i|s_i, z_i)$, on the other hand, is supposed to denotes the probability of z_i being assigned to the coarse-grained category y_i when its fine-grained category s_i is specified, which is formulated as:

$$q(y{=}j|s_i, z_i) = \frac{e^{-\frac{1}{2}\left(\|z_i-\bar{\mu}_j\|_2^2+\|\mu_{s_i}-\bar{\mu}_j\|_2^2\right)/\tau_y}}{\sum_{j'=1}^{M} e^{-\frac{1}{2}\left(\|z_i-\bar{\mu}_{j'}\|_2^2+\|\mu_{s_i}-\bar{\mu}_{j'}\|_2^2\right)/\tau_y}}, \quad (12)$$

where τ_y denotes the temperature. Referring to the reparameterization trick for categorical distribution in (Jang et al., 2016), s_i and y_i are represented as $s_i = \arg\max_s [q(s|z_i) + \psi_i^s]$ and $y_i = \arg\max_y [q(y|s_i, z_i) + \psi_i^y]$, where ψ_i^s and ψ_i^y are drawn from the Gumbel distribution $Gumbel(0, 1)$. The calculation of $\mathbb{KL}(q(s_i|z_i)\|p(s_i|y_i))$ and $\mathbb{KL}(q(y_i|s_i, z_i)\|p(y_i))$ are introduced in appendix A.4 and A.5. It is also worth noting that the inferred s_i and y_i can serve as hierarchical pseudo labels for each z_i.

Thanks to employing the hierarchical distribution $p(s, y) = p(s|y)p(y)$ in (3) as the prior, as we maximize the lower bound $\mathcal{L}_{\text{ELBO}}$ in (7), the fine-grained and coarse-grained prototypes $\{\mu_s\}_{s=1}^K$ and $\{\bar{\mu}_y\}_{y=1}^M$ will be driven to align with the true semantic hierarchical structure hidden in the documents, as illustrated in Figure 2.

3.3 Promoting the Hierarchical Structure via Contrastive Learning

Ideally, maximizing the lower bound $\mathcal{L}_{\text{ELBO}}$ should be able to discover the underlying semantic hierarchy. But given the difficulties of this task, simply maximizing the lower bound $\mathcal{L}_{\text{ELBO}}$ is often not enough. In this section, we propose to promote the semantic hierarchies in the learned representations. It is noticed that the semantic information in documents is often largely hierarchical. Thus, if the latent representations z_i are encouraged to retain as much semantic information as possible, they should largely align with the underlying hierarchical semantic structure by nature. Therefore, in addition to optimizing the bound $\mathcal{L}_{\text{ELBO}}$, we force the latent representations z_i to preserve as much semantic information as possible via contrastive learning, in which the hierarchical pseudo labels discovered by the posteriors $q(s_i|z_i)$ and $q(y_i|s_i, z_i)$ are leveraged to strengthen the hierarchical structure.

Specifically, given a document x_i, the augmentation counterpart x_i^+ is first obtained with dropout of BERT model as has been proposed in SimCSE (Gao et al., 2021). Coarse-grained pseudo labels are first utilized in sample selection: Denote by $Z^-(i) = \left\{z_j, z_j^+\right\}_{j=1,\cdots,n,j\neq i}$ the negative set for a sample x_i, where n is the number of samples within a batch. $Z^-(i)$ can be divided into $Z^-_{\text{diff}}(i)$ and $Z^-_{\text{same}}(i)$ according to the pseudo labels as:

$$Z^-_{\text{diff}}(i) = \left\{z_j \in Z^-(i) : y_j \neq y_i\right\} \qquad (13)$$

$$Z^-_{\text{same}}(i) = \left\{z_j \in Z^-(i) : y_j = y_i\right\}. \qquad (14)$$

For latent representations in $Z^-_{\text{diff}}(i)$, their semantic differences with anchor z_i are enlarged with the conventional contrastive loss (Chen et al., 2020):

$$\mathcal{L}_{\text{diff}} = \sum_{i=1}^{N} \log \frac{-e^{\text{sim}(z_i, z_i^+)}}{\sum_{z_j \in \left\{\{z_i\} \cup Z^-_{\text{diff}(i)}\right\}} e^{\text{sim}(z_i, z_j)}}, \qquad (15)$$

where $\text{sim}(\cdot, \cdot)$ denotes a similarity function. We hereby apply cosine similarity which makes $\text{sim}(a, b) = \frac{\cos(a,b)}{\tau}$ where τ is temperature.

The representations in $Z^-_{\text{same}}(i)$, on the other hands, have more similarities with z_i since they share the same coarse-grained labels, which means pushing them away directly may deteriorate the hash model by destroying the coarse-grained clusters in latent space. Therefore, we turn to angular normalization (Bukchin et al., 2021), which transforms a raw representation into a unit vector a_i which indicates the angle between the original representation and its corresponding prototype (*i.e.* $\bar{\mu}_y$).

$$a_i = \frac{\frac{z_i}{\|z_i\|} - \frac{\bar{\mu}_{y_i}}{\|\bar{\mu}_{y_i}\|}}{\left\|\frac{z_i}{\|z_i\|} - \frac{\bar{\mu}_{y_i}}{\|\bar{\mu}_{y_i}\|}\right\|}. \qquad (16)$$

We denote $A^-_{\text{same}}(i)$ as $Z^-_{\text{same}}(i)$ after angular normalization:

$$\mathcal{L}_{\text{same}} = \sum_{i=1}^{N} \log \frac{-e^{\text{sim}(a_i, a_i^+)}}{\sum_{a_j \in \left\{\{a_i\} \cup A^-_{\text{same}(i)}\right\}} e^{\text{sim}(a_i, a_j)}}. \qquad (17)$$

$\mathcal{L}_{\text{diff}}$ and $\mathcal{L}_{\text{same}}$ are combined to form a coarse-grained contrastive constraint as $\mathcal{L}_{\text{coarse}} = \mathcal{L}_{\text{diff}} + \mathcal{L}_{\text{same}}$, where $\mathcal{L}_{\text{diff}}$ directly pushes negative samples with different coarse labels away from the anchor z_i to strengthen the preserved coarse-grained semantics, and $\mathcal{L}_{\text{same}}$ operates on angular vectors of the same coarse labels, facilitating the preservation of fine-grained semantics within the coarse-grained categories.

To further refine fine-grained semantics, we exploit $q(s_i|z_i)$ to define the probability distribution of z_i being assigned to fine-grained pseudo labels as $Q_i^s = [q(s = 1|z_i), \cdots, q(s = K|z_i)] \in [0, 1]^K$, which can serve as representation to reflect fine-grained semantics of sample. A fine-grained semantic constraint is formulated as:

$$\mathcal{L}_{\text{fine}} = -\sum_{i=1}^{N} (\log\langle Q_i^s \cdot Q_{i+}^s\rangle) - \hat{I}(S; Z) \qquad (18)$$

where $\langle a \cdot b\rangle$ denotes dot product; $\hat{I}(S; Z)$ denotes the mutual information (MI) between the fine-grained pseudo labels S and latent variables Z, which is estimated as:

$$\hat{I}(S; Z) = \tilde{H}(\frac{1}{N} \sum_{i=1}^{N} Q_i^s) - \eta \sum_{i=1}^{N} \tilde{H}(Q_i^s), \qquad (19)$$

where $\tilde{H}(Q_i^s)$ denotes the entropy over the probability distribution of Q_i^s; $\frac{1}{N}\sum_{i=1}^{N} Q_i^s$ is used to estimate the marginal distribution of S; η is the hyperparameter to adjust the relative importance of the two terms. $\hat{I}(S; Z)$ is able to avoid model collapse where all samples are assigned to the same fine-grained category.

Overall Loss The final loss of the hierarchical generative model is formulated as:

$$\mathcal{L} = -\mathcal{L}_{\text{ELBO}} + \alpha \cdot \mathcal{L}_{\text{coarse}} + \beta \cdot \mathcal{L}_{\text{fine}}. \qquad (20)$$

After the model is trained, the mean value of latent variable z_i (*i.e.* $\tilde{\mu}_i$) is utilized to obtain the final hash code with $b_i = \text{sign}(\text{sigmoid}(\tilde{\mu}_i) - 0.5)$. With the model implicitly introducing hierarchical semantics into latent representations z_i, it acquires the ability to generate hash codes with hierarchical semantics.

3.4 Further Improving with Hierarchical Self-Labeling

With the optimization of $\mathcal{L}$ in Equation 20 , the model is expected to produce more reliable pseudo labels, therefore it's beneficial to strengthen the influence of pseudo labels to further improve the model. It is observed that high-confidence pseudo labels perform well in many different tasks including clustering (Van Gansbeke et al., 2020) and hashing (Song et al., 2023). Inspired of this, we apply a hierarchical self-labeling module to further train our model in a supervised way, in which pseudo labels with extremely high confidence are leveraged as ground truth. Specifically, thresholds γ_y and γ_s are set for filtering the samples with confident coarse/fine-grained pseudo labels:

$$\mathbb{C}_y = \{i \in \{1, \cdots, N\} : q(y_i|s_i, z_i) > \gamma_y\} \quad (21)$$

$$\mathbb{C}_s = \{i \in \{1, \cdots, N\} : q(s_i|z_i) > \gamma_s\}. \quad (22)$$

Then, the pseudo labels of selected samples are leveraged as supervision signals to guide the predictions. To alleviate overfitting, we use the probability of samples' augmentation counterparts during the self-labeling process, thus the loss is formulated as:

$$\begin{aligned}\mathcal{L}_{\text{self}} = &- \frac{1}{|\mathbb{C}_s|}\sum_{i\in\mathbb{C}_s} \text{XE}(Q^s_{i+}, s_i) \\ &- \frac{1}{|\mathbb{C}_y|}\sum_{i\in\mathbb{C}_y} \text{XE}(Q^y_{i+}, y_i),\end{aligned} \quad (23)$$

where XE denotes the cross-entropy loss; $Q_i^y = [q(y = 1|s_i, z_i), \cdots, q(y = M|s_i, z_i)] \in [0, 1]^M$ denotes the coarse-grained probability.

Due to pseudo labels relying on latent variables, the minimization of $\mathcal{L}_{\text{self}}$ in Equation 23 promotes the latent representations to form more compact and separable hierarchical clusters, consequently improving the quality of hash codes.

4 Experiment[1]

4.1 Experimental setups

Dataset The model is evaluated on three public datasets, including:

- **NYT** (Tao et al., 2018) contains news articles published by The New York Times.
- **DBpedia** (Lehmann et al., 2015) contains abstract of articles from Wikipedia.
- **AGNews** (Zhang et al., 2015) consists of news gathered from academic news search engines.

All of the above datasets are in English, and are randomly split into training, validation and test sets with the proportion of 8:1:1. The detailed statistics of the datasets can be found in Table 1.

Dataset	Coarse	Fine	DocNum	AvgLen
NYT	5	26	13081	648.13
DBpedia	9	70	50000	103.37
AGNews	4	NAN	127600	31.59

Table 1: Statistics of datasets. Coarse: number of coarse-grained categories; Fine: number of fine-grained categories; DocNum: number of documents; AvgLen: average length of documents.

Baselines We compare our model with the following unsupervised deep semantic hashing methods: VDSH (Chaidaroon and Fang, 2017), NASH (Shen et al., 2018), BMSH (Dong et al., 2019), CorrSH (Zheng et al., 2020), WISH (Ye et al., 2020), PairRec (Hansen et al., 2020), SNUH (Ou et al., 2021a), SMASH (He et al., 2023), DHIM (Ou et al., 2021b) and MICPQ (Qiu et al., 2022). They are all experimented on both tf-idf features and BERT features except for SMASH, DHIM and MICPQ. The performances of VDSH, NASH, BMSH, CorrSH and WISH using tf-idf features in NYT and AGNews are quoted from DHIM (Ou et al., 2021b). Others are implemented in our environment and corresponding performances are obtained.

Evaluation Metric We evaluate the methods with the top-100 retrieval precision (P@100) following previous works. Top-100 most similar documents are retrieved for every query documents in the testing set based on the hamming distance between corresponding hashing codes. A retrieved document is considered relevant to the query document if they share the same label. Finally, the retrieval precision averaged over all test documents is reported. For all methods, we average the P@100 over 5 random runs.

4.2 Implementation details

We use the [CLS] token of pretrained BERT model (Devlin et al., 2019) which is finetuned with unsupervised contrastive loss on three datasets respectively as input document feature. Following DHIM (Ou et al., 2021b), we fix the BERT model while only training the newly proposed part. The BERT

[1] Our code is available at: `https://github.com/Emily-zero/HierHash`

method	**NYT**				**AGNews**				**DBpedia**			
	16	32	64	128	16	32	64	128	16	32	64	128
using tf-idf features												
VDSH	68.77	68.77	75.01	78.49	67.32	67.42	72.70	73.86	44.97	54.44	60.84	64.38
NASH	74.87	75.52	75.08	73.01	65.74	69.34	72.72	74.33	54.04	60.18	63.50	65.85
WISH	70.15	70.03	64.48	68.94	74.53	74.79	75.05	72.70	50.79	62.79	66.37	65.32
BMSH	74.02	76.38	76.88	77.63	74.09	76.03	76.09	73.56	54.60	59.68	60.78	63.03
CorrSH	75.43	77.61	77.24	78.39	76.20	76.45	76.61	77.67	58.45	64.80	69.17	70.93
PairRec	75.42	78.56	79.63	80.75	76.82	78.31	79.79	80.18	41.03	46.48	49.60	52.11
SNUH	76.22	79.33	80.66	81.64	79.66	81.08	81.45	80.56	54.90	65.43	69.15	68.56
SMASH	77.40	73.86	79.05	81.38	73.12	70.16	71.62	76.34	55.65	59.16	61.87	62.82
using BERT features												
VDSH	76.72	78.19	79.54	80.75	77.04	78.93	79.61	80.75	59.32	70.86	76.02	78.88
NASH	76.00	78.73	79.42	80.79	77.17	78.19	78.97	79.50	58.99	70.97	76.06	77.83
WISH	77.29	77.27	79.76	72.79	78.00	79.35	79.74	80.58	66.12	73.74	74.77	42.35
BMSH	76.92	79.51	80.24	80.51	66.77	69.61	71.99	73.16	56.33	68.72	73.81	78.13
CorrSH	77.90	79.27	80.03	80.47	77.82	79.04	79.92	79.91	57.93	68.50	72.54	75.32
PairRec	77.44	79.73	81.18	82.02	80.42	81.50	82.09	82.22	55.33	66.78	75.00	77.18
SNUH	66.08	76.32	82.25	82.53	76.27	79.37	80.05	81.69	54.51	64.18	67.27	70.88
DHIM	79.69	80.55	79.77	79.09	78.23	79.17	78.88	79.86	49.98	63.58	69.25	52.37
MICPQ	80.86	80.67	82.20	81.99	80.19	81.65	82.08	82.04	68.51	75.86	76.86	78.92
HierHash	**81.54**	**82.42**	**84.47**	**85.28**	**81.55**	**82.38**	**82.99**	**83.45**	**69.10**	**81.14**	**84.16**	**85.30**

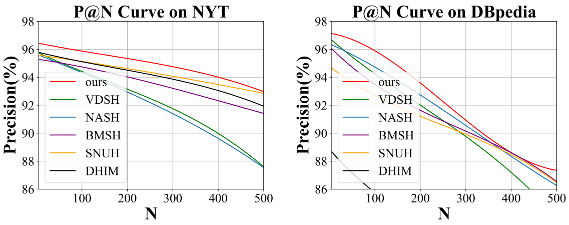
Table 2: P@100 on three datasets with different numbers of bits in unsupervised document hashing.

feature used by baselines are exactly the same as our proposed model in the reported experiments. For the tf-idf features of baselines, we obtain them with scikit-learn package (Pedregosa et al., 2011).

g_{ϕ_1} and g_{ϕ_2} in Equation 9 and f_θ in Equation 6 are both two-layer feed forward network whose hidden layer are fixed to 1024 and 128, with a ReLU as activation function. The learning rates are fixed to 5e-4 for NYT and DBpedia and 1e-4 for AGNews. The number of coarse-grained categories M is set as the ground truth class number in three datasets while the number of fine-grained categories K are 100 for NYT and DBpedia, 20 for AGNews. The τ, τ_y, τ_s and η are fixed to 0.3,0.2,1.0 and 0.1 respectively for all three datasets. $\alpha \in \{1, 2, 5, 10\}$ and $\beta \in \{0.1, 0.5, 1\}$ are tuned according to validation sets. The model is implemented with PyTorch and transformers (Wolf et al., 2020), and Adam optimizer (Kingma and Ba, 2015) is applied for optimization. We warm up the model with Equation 20 for 15 epochs and reinitialize $\{\mu_s\}_{s=1}^{K}$ by appying k-means on the training set of data. $\{\bar{\mu}_y\}_{y=1}^{M}$ is reinitialized by applying k-means on $\{\mu_s\}_{s=1}^{K}$.

4.3 Experimental Result

Overall Performance The precision@100 results on three public benchmarks with encoding length of 16,32,64 and 128 are demonstrated in Table 2. It is obvious that HierHash yields the best performance in all cases. Specifically, HierHash outperforms the second best performance by 0.68%, 1.75%, 2.22% and 2.75% for NYT and by 0.59%, 5.28%, 7.30% and 6.38% for DBpedia, which proves the superiority of our proposed model in hierarchical datasets. Despite this, HierHash achieves better performance in flat dataset AGNews as well, with a performance gain of 1.13%, 0.73%, 0.90% and 1.23% respectively compared to the second best performance on the four encoding length. The significant improvements serve as a solid prove that the modeling of hierarchical structure of semantics is generalizable among datasets, regardless of their provided labels, further confirming the effectiveness of our method and the benefits of preserving hierarchical semantics.



(a) NYT dataset (b) DBpedia dataset

Figure 3: P@N Curve of HierHash and compared methods on coarse-grained labels of two hierarchical datasets with 128-bit hash code length.

We also demonstrate the P@N curves of coarse-grained categories of two hierarchical datasets with 128-bit code length, which are shown in Figure 3. It can be found that the coarse-grained precision of our proposed method is higher than other compared methods at almost all of the numbers of returned results, and the precision drops more slowly than other methods, which proves the effectiveness on hierarchical retrieval of HierHash. The complete P@100 results can be found in appendix C.

Ablation Study We conduct ablation studies on variants of HierHash to understand the effect of major components of it. (i) Base is the generative model without hierarchical categories, the implementation of which is the same as VDSH model that utilizes BERT embedding. (ii)w/hier is the proposed hierarchical generative model without further promoting the hierarchical structure. (iii) w/$\mathcal{L}_{\text{con}}$ is to apply contrastive learning on hierarchical generative model, with an overall loss of $\mathcal{L}$ in Equation 20. (iv) w/$\mathcal{L}_{\text{self}}$ is our final hierarchical generative model which comprised of contrastive learning and hierarchical self-labeling module. We list the results on three datasets with different code length in Table 3.

Ablation		16	32	64	128
NYT	Base	76.72	78.19	79.54	80.75
	w/hier	81.38	81.67	84.14	85.22
	w/$\mathcal{L}_{\text{con}}$	81.50	83.22	85.85	85.62
	w/$\mathcal{L}_{\text{self}}$	**81.64**	**83.42**	**85.89**	**85.72**
AGNews	Base	77.04	78.93	79.61	80.75
	w/hier	81.16	81.82	82.36	83.04
	w/$\mathcal{L}_{\text{con}}$	80.59	81.72	82.32	83.30
	w/$\mathcal{L}_{\text{self}}$	**81.55**	**82.38**	**82.99**	**83.45**
DBpedia	Base	59.32	70.86	76.02	78.88
	w/hier	64.05	73.08	77.49	79.48
	w/$\mathcal{L}_{\text{con}}$	68.83	80.05	82.93	84.26
	w/$\mathcal{L}_{\text{self}}$	**69.72**	**81.14**	**84.19**	**85.30**

Table 3: The P@100 of variant models on three datasets.

As is shown in row one and row two, the hierarchical structure brings in performance gain for all encoding length in all datasets, proving the benefits of introducing hierarchical structure into the generative model. Then compare row two and row three, the significant performance gain in NYT and DBpedia indicates the effectiveness of further promoting hierarchical structure via contrastive learning. In row four, the hierarchical self-labeling module further improves the performances, which proves the importance of strengthening the supervision from hierarchical pseudo labels.

Hyperparameter Analysis We mainly evaluate the model performance on 64-bit hash codes with different values of temperature τ,τ_s and τ_y, which serve as important hyperparameters for the basic hierarchical generative model and the hierarchical contrastive learning module. As is shown in Figure 4, HierHash is not sensitive to τ_s and τ_y as the P@100 varies a little with different values. DBPedia is sensitive to τ while the other two datasets witness performance drop only when τ is too large. In addition, we explore the influence of fine-grained category number K on NYT where we vary the setting of K from $K \in \{20, 50, 100, 150, 200\}$. The results show that NYT is relatively stable when K varies, indicating the robustness of our model.

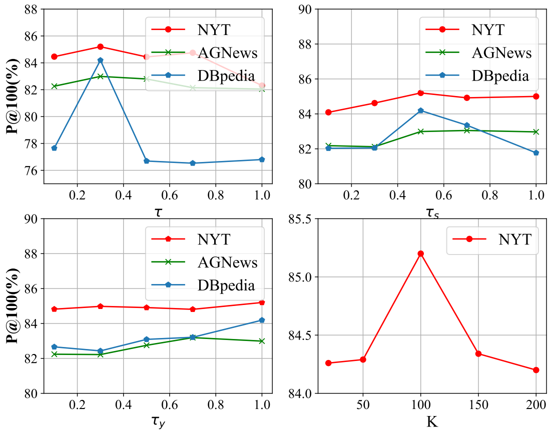


Figure 4: Impact of three temperature coefficients and fine cluster number K with 64-bit hash codes.

Case Study In table 4, we depict two concrete cases of similarity search with 64-bits hash code on DBpedia dataset for the intuitive understanding of HierHash. It's obvious that when the hamming distance is relatively small, the retrieved documents share the same coarse-grained categories and fine-grained categories with the query. With the increase of hamming distance, the fine-grained and coarse-grained categories of the retrieved documents change subsequently, demonstrating that the hamming distance can effectively measure the relevance of documents and the hash codes generated by our model indeed preserve hierarchical semantics of documents.

Visualization In Figure 5, we display the t-SNE visualization (van der Maaten and Hinton, 2008) of hash codes on DBpedia with different variants of 64

Distance	Content	Coarse Category	Fine Category
query	... Dam is a rock-fill embankment dam...	Place	Infrastructure
5	Inkachaka Dam is a dam in Bolivia situated in	Place	Infrastructure
10	The Muscoot Reservoir is a reservoir...	Place	Infrastructure
20	Puente Viejo is ... of three bridges...	Place	Route Of Transportation
30	ACapra is an American musical group ...	Agent	Group
query	Turbonilla miona is a species of sea snail...	Species	Animal
5	Papillifera deburghiae, is a species of ... land snail...	Species	Animal
10	Neotama is a genus of tree trunk spiders ...	Species	Animal
20	Sphaerodictyaceae is a family of green alga...	Species	Plant
30	Paul J. Wasicka ... is a professional poker player...	Agent	Athlete

Table 4: The documents with Hamming distances of 5, 10,20 and 30 to the query of the 64-bit hash codes on the DBpedia dataset.

bits. The color of data points indicates the coarse-grained category the samples belong. The figure shows that HierHash is able to form finer-grained clusters within coarse-grained clusters, proving its ability to preserve hierarchical semantics.

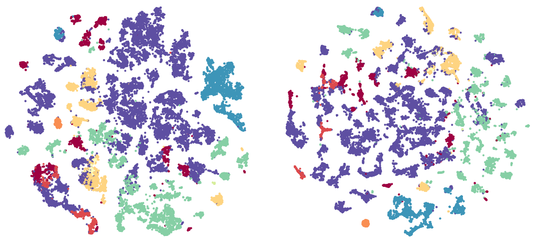

(a) Base (b) HierHash

Figure 5: visualization

5 Conclusion

In this paper, to learn hash codes with hierarchical semantics unsupervisedly, we propose HierHash: a hierarchical generative model which introduces multi-grained prototypes to integrate the hierarchical structure of semantics into hash model. Furthermore, to promote the learned hierarchical structure, we leverage hierarchical pseudo-labels produced during the VAE optimization with contrastive-based methods, which refine the semantics preserved in hash codes. Experimental results show that HierHash significantly outperforms existing baselines on both hierarchical datasets and flat datasets.

Limitations

Due to the Gaussian assumption in generative models, the hash codes are binarized from continuous latent variables z. The two-stage training procedure would compromise the performance. Therefore, our future work will focus on designing an end-to-end generative hashing with hierarchical structure.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 62276280, U1811264), Guangzhou Science and Technology Planning Project (No. 2024A04J9967).

References

Guy Bukchin, Eli Schwartz, Kate Saenko, Ori Shahar, Rogerio Feris, Raja Giryes, and Leonid Karlinsky. 2021. Fine-grained angular contrastive learning with coarse labels. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8726–8736.

Suthee Chaidaroon and Yi Fang. 2017. Variational deep semantic hashing for text documents. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 75–84.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709.

Zhuo Chen, Ruizhou Ding, Ting-Wu Chin, and Diana Marculescu. 2018. Understanding the impact of label granularity on cnn-based image classification. *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 895–904.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Wei Dong, Qinliang Su, Dinghan Shen, and Changyou Chen. 2019. Document hashing with mixture-prior generative models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5226–5235.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 6894–6910. Association for Computational Linguistics (ACL).

Jia-Nan Guo, Xian-Ling Mao, Wei Wei, and Heyan Huang. 2023. Intra-category aware hierarchical supervised document hashing. *IEEE Transactions on Knowledge and Data Engineering*, 35(6):6003–6013.

Yuanfan Guo, Minghao Xu, Jiawen Li, Bingbing Ni, Xuanyu Zhu, Zhenbang Sun, and Yi Xu. 2022. Hcsc: Hierarchical contrastive selective coding. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9696–9705.

Casper Hansen, Christian Hansen, Jakob Grue Simonsen, Stephen Alstrup, and Christina Lioma. 2020. Unsupervised semantic hashing with pairwise reconstruction. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Liyang He, Zhenya Huang, Enhong Chen, Qi Liu, Shiwei Tong, Hao Wang, Defu Lian, and Shijin Wang. 2023. An efficient and robust semantic hashing framework for similar text search. *ACM Trans. Inf. Syst.*, 41(4).

Eric Jang, Shixiang Shane Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *ArXiv*, abs/1611.01144.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational bayes. *CoRR*, abs/1312.6114.

Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Knowledge Discovery and Data Mining*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven C. H. Hoi. 2020. Prototypical contrastive learning of unsupervised representations. *ArXiv*, abs/2005.04966.

Jingchao Ni, Wei Cheng, Zhengzhang Chen, Takayoshi Asakura, Tomoya Soma, Sho Kato, and Haifeng Chen. 2021. Superclass-conditional gaussian mixture model for learning fine-grained embeddings. In *International Conference on Learning Representations*.

Zijing Ou, Qinliang Su, Jianxing Yu, Bang Liu, Jingwen Wang, Ruihui Zhao, Changyou Chen, and Yefeng Zheng. 2021a. Integrating semantics and neighborhood information with graph-driven generative models for document retrieval. *arXiv preprint arXiv:2105.13066*.

Zijing Ou, Qinliang Su, Jianxing Yu, Ruihui Zhao, Yefeng Zheng, and Bang Liu. 2021b. Refining bert embeddings for document hashing via mutual information maximization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2360–2369.

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.

Zexuan Qiu, Qinliang Su, Zijing Ou, Jianxing Yu, and Changyou Chen. 2021. Unsupervised hashing with contrastive information bottleneck. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 959–965. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Zexuan Qiu, Qinliang Su, Jianxing Yu, and Shijing Si. 2022. Efficient document retrieval by end-to-end refining and quantizing BERT embedding with contrastive product quantization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 853–863. Association for Computational Linguistics.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China. PMLR.

Ruslan Salakhutdinov and Geoffrey Hinton. 2009. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978.

Dinghan Shen, Qinliang Su, Paidamoyo Chapfuwa, Wenlin Wang, Guoyin Wang, Ricardo Henao, and Lawrence Carin. 2018. Nash: Toward end-to-end neural architecture for generative semantic hashing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2041–2050.

Zhenpeng Song, Qinliang Su, and Jiayang Chen. 2023. Unsupervised hashing with contrastive learning by exploiting similarity knowledge and hidden structure of data. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, page 6350–6358, New York, NY, USA. Association for Computing Machinery.

Richard P. Stanley. 2011. *Enumerative Combinatorics: Volume 1*, 2nd edition. Cambridge University Press, USA.

Benno Stein, Sven Meyer zu Eissen, and Martin Potthast. 2007. Strategies for retrieving plagiarized documents. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Changchang Sun, Xuemeng Song, Fuli Feng, Wayne Xin Zhao, Hao Zhang, and Liqiang Nie. 2019. Supervised hierarchical cross-modal hashing. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Fangbo Tao, Chao Zhang, Xiusi Chen, Meng Jiang, Tim Hanratty, Lance Kaplan, and Jiawei Han. 2018. Doc2cube: Allocating documents to text cube without labeled data. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1260–1265. IEEE.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. 2020. Scan: Learning to classify images without labels. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X*, page 268–285, Berlin, Heidelberg. Springer-Verlag.

Dan Wang, Heyan Huang, Chi Lu, Bo-Si Feng, Liqiang Nie, Guihua Wen, and Xian-Ling Mao. 2017. Supervised deep hashing for hierarchical labeled data. In *AAAI Conference on Artificial Intelligence*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Minghao Xu, Yuanfan Guo, Xuanyu Zhu, Jiawen Li, Zhenbang Sun, Jiangtao Tang, Yi Xu, and Bingbing Ni. 2022. Hirl: A general framework for hierarchical image representation learning. *ArXiv*, abs/2205.13159.

Jinhai Yang, Han Yang, and Lin Chen. 2020. Towards cross-granularity few-shot learning: Coarse-to-fine pseudo-labeling with visual-semantic meta-embedding. *Proceedings of the 29th ACM International Conference on Multimedia*.

Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. 2020. Unsupervised few-bits semantic hashing with implicit topics modeling. In *EMNLP (Findings)*, volume 20, pages 2566–2575. Association for Computational Linguistics (ACL).

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Lin Zheng, Qinliang Su, Dinghan Shen, and Changyou Chen. 2020. Generative semantic hashing enhanced via boltzmann machines. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 777–788.

A VAE derivation

A.1 ELBO derivation

By applying Jensen inequality, ELBO(Evidence Lower Bound) of $\log p(x_i)$ can be derived:

$$\begin{aligned}\log p(x_i) &= \log \int_{z_i} \sum_{s_i=1}^{K} \sum_{y_i=1}^{M} p(x_i, y_i, s_i, z_i) dz_i \\ &= \log \int_{z_i} \sum_{s_i=1}^{K} \sum_{y_i=1}^{M} q_\phi(y_i, z_i, s_i|x_i) \\ &\quad \cdot \frac{p(x_i, y_i, s_i, z_i)}{q_\phi(y_i, z_i, s_i|x_i)} dz_i \\ &= \log \left\{ \mathbb{E}_{q_\phi(y_i,z_i,s_i|x_i)} \left[\frac{p(x_i, y_i, s_i, z_i)}{q_\phi(y_i, z_i, s_i|x_i)} \right] \right\} \\ (\text{Jenson}) &\geq \mathbb{E}_{q_\phi(y_i,z_i,s_i|x_i)} \left[\log \frac{p(x_i, y_i, s_i, z_i)}{q_\phi(y_i, z_i, s_i|x_i)} \right].\end{aligned} \tag{24}$$

Then,substituting $q_\phi(y_i, z_i, s_i|x_i)$ in denominator of Equation 24 with Equation 8, we can further write ELBO in the following way:

$$
\begin{aligned}
&\mathcal{L}_{\text{ELBO}} \\
&= \mathbb{E}_{q_\phi(y_i, z_i, s_i|x_i)}\left[\log \frac{p_\theta(x_i|z_i)\cdot p(z_i|s_i)\cdot p(s_i, y_i)}{q_\phi(y_i, z_i, s_i|x_i)}\right] \\
&= \mathbb{E}_{q_\phi(z_i|x_i)}[\log p_\theta(x_i|z_i)] + \mathbb{E}_{q_\phi(z_i|x_i)}\left[\log \frac{p(z_i|s_i)}{q_\phi(z_i|x_i)}\right] \\
&\quad + \mathbb{E}_{q_\phi(s_i|z_i)}\left[\log \frac{p(s_i)}{q_\phi(s_i|z_i)}\right] \\
&\quad + \mathbb{E}_{q_\phi(y_i|s_i)}\left[\log \frac{p(y_i|s_i)}{q_\phi(y_i|s_i, x_i)}\right] \\
&= \mathbb{E}_{q_\phi(z_i|x_i)} p_\theta(x_i|z_i) \\
&\quad - \mathbb{KL}(q_\phi(z_i|x_i)\|p(z_i|s_i)) \\
&\quad - \mathbb{KL}(q_\phi(s_i|z_i)\|p(s_i)) \\
&\quad - \mathbb{KL}(q_\phi(y_i|s_i)\|p(y_i|s_i)).
\end{aligned}
\tag{25}
$$

A.2 $\mathbb{E}_{q_\phi(z_i|x_i)} \log p_\theta(x_i|z_i)$

$p(x_i|z_i)$ is a multivariant Gaussian distribution, we have:

$$
\begin{aligned}
&\mathbb{E}_{q_\phi(z_i|x_i)} \log p_\theta(x_i|z_i) = \\
&\mathbb{E}_{q_\phi(z_i|x_i)}\left[\log\left[\frac{\exp -\frac{(x_i-\mu_{x_i})^\top(\sigma_{x_i}^2 I)^{-1}(x_i-\mu_{x_i})}{2}}{\sqrt{(2\pi)^d|\sigma_{x_i}^2 I|}}\right]\right],
\end{aligned}
\tag{26}
$$

where d is the length of input document feature x.

Ignore the constant value and conduct Monte Carlo sampling, the expectation can be approximated by the following equation where N^m is the number of Monte Carlo samples:

$$
\begin{aligned}
&\mathbb{E}_{q_\phi(z_i|x_i)} p_\theta(x_i|z_i) = \\
&\sum_{m=1}^{N^m} \log\left[\frac{\exp -\frac{(x_i-\mu_{x_i^m})^\top(\sigma_{x_i^m}^2 I)^{-1}(x_i-\mu_{x_i^m})}{2}}{\sqrt{(2\pi)^d|\sigma_{x_i^m}^2 I|}}\right],
\end{aligned}
\tag{27}
$$

where $\mu_{x_i^m}$ and $\sigma_{x_i^m}$ denotes the m$^{\text{th}}$ Monte Carlo sample of x_i. For z_i which is sampled from a gaussian distribution, it is reparameterized as $z_i = \tilde{\mu}_i + \xi_i \cdot \tilde{\sigma}_i$ with $\xi_i \sim \mathcal{N}(0, I)$.

A.3 $\mathbb{KL}(q_\phi(z_i|x_i)\|p(z_i|s_i))$

For the second term of $\mathcal{L}_{\text{ELBO}}$, that is $\mathbb{KL}(q_\phi(z_i|x_i)\|p(z_i|s_i))$, it can be easily calculated as the KL divergence between two Gaussian distribution:

$$
-\frac{1}{2}\sum_{j=1}^{L}\left[\log\frac{\tilde{\sigma}_{ij}^2}{{\sigma_{s_i}^j}^2} - \frac{\tilde{\sigma}_{ij}^2}{{\sigma_{s_i}^j}^2} + 1 - \frac{(\tilde{\mu}_{ij} - \mu_{s_i}^j)^2}{{\sigma_{s_i}^j}^2}\right],
\tag{28}
$$

where μ_s, σ_s are learnable parameters varied with different s. $\tilde{\sigma}_{ij}$ and $\tilde{\mu}_{ij}$ indicate the j$^{\text{th}}$ element of $\tilde{\sigma}_i$ and $\tilde{\mu}_i$, respectively.

A.4 $\mathbb{KL}(q_\phi(s_i|z_i)\|p(s_i|y_i))$

To calculate the third term in $\mathcal{L}_{\text{ELBO}}$, we fix the softmax temperature as τ_s and conduct L2 normalization to μ_{s_i} and z_i:

$$
\begin{aligned}
\exp(-\|z_i - s_i\|^2/\tau_s) &= \exp -\frac{(z_i-\mu_{s_i})^\top(z_i-\mu_{s_i})}{\tau_s} \\
&= \exp -\frac{z_i^2 + \mu_{s_i}^2 - 2z_i^\top\mu_{s_i}}{\tau_s} \\
&= \exp -\frac{z_i^2 + \mu_{s_i}^2 - 2z_i^\top\mu_{s_i}}{\tau_s} \\
&= \exp -\frac{2 - 2z_i^\top\mu_{s_i}}{\tau_s}.
\end{aligned}
\tag{29}
$$

Therefore,$q(s_i|z_i)$ can be written as:

$$
q(s_i|z_i) = \frac{\exp(2z_i^\top \cdot \mu_{s_i}/\tau_s)}{\sum_{j=1}^K \exp(z_i^\top \cdot \mu_j/\tau_s)}.
\tag{30}
$$

$\mathbb{KL}(q_\phi(s_i|z_i)\|p(s_i|y_i))$ is further calculated as:

$$
\begin{aligned}
&\mathbb{KL}(q_\phi(s_i|z_i)\|p(s_i|y_i)) = \mathbb{E}_{q_(s_i|z_i)}\left[\log\frac{p(s_i|y_i)}{q(s_i|z_i)}\right] \\
&= \left[-\log\left[\frac{1}{p(s_i|z_i)}\cdot\frac{\exp(2z_i^\top\cdot\mu_{s_i}/\tau_s)}{\sum_{j=1}^K\exp(z_i^\top\cdot\mu_j/\tau_s)}\right]\right],
\end{aligned}
\tag{31}
$$

For the stability of training, we precalculate $p(s_i|y_i)$ at the beginning of each epoch, thus $p(s_i|y_i)$ is a constant once the s_i and y_i is determined.

A.5 $\mathbb{KL}(q_\phi(y_i|s_i, z_i)\|p(y_i))$

Similar to $\mathbb{KL}(q_\phi(s_i|z_i)\|p(s_i|y_i))$, we conduct L2 normalization to y_i,s_i and z_i, with the temperature fixed to τ_y:

$$
\begin{aligned}
&\exp\left[\frac{-\frac{1}{2}(\|z_i - \bar{\mu_{y_i}}\|^2 + \frac{1}{2}\|\mu_{s_i} - \bar{\mu_{y_i}}\|^2)}{\tau_y}\right] \\
&= \exp\left[\frac{-(2 - z_i^\top\bar{\mu}_{y_i} - \mu_{s_i}^\top\bar{\mu}_{y_i})}{\tau_y}\right].
\end{aligned}
\tag{32}
$$

Therefore,$q(y_i|s_i, z_i)$ can be written as:

$$
q(y_i|s_i, z_i) = \frac{\exp\left[(z_i^\top\bar{\mu}_{y_i} + \mu_{s_i}^\top\bar{\mu}_{y_i})/\tau_y\right]}{\sum_{j=1}^M\exp\left[(z_i^\top\bar{\mu}_j + \mu_{s_i}^\top\bar{\mu}_j)/\tau_y\right]}.
\tag{33}
$$

method	NYT				DBpedia			
	16	32	64	128	16	32	64	128
using tf-idf features								
VDSH	89.43	91.79	92.59	91.53	74.47	79.41	82.73	84.78
NASH	92.90	92.74	94.41	94.04	79.59	83.20	84.36	85.25
WISH	92.39	93.24	94.31	94.67	78.94	85.12	87.04	87.32
BMSH	94.00	91.76	92.44	92.45	80.34	82.74	81.78	83.03
CorrSH	92.78	94.55	94.26	94.30	82.20	85.39	87.59	88.24
PairRec	89.69	91.00	91.49	92.73	55.14	56.42	58.25	60.93
SNUH	81.02	93.15	94.85	94.78	81.51	85.52	87.17	87.05
SMASH	94.14	91.95	91.93	93.49	80.36	81.39	83.04	83.41
using BERT features								
VDSH	93.30	93.58	94.13	94.55	87.21	91.04	93.05	93.54
NASH	93.61	93.90	94.39	94.88	87.81	92.42	93.13	94.32
WISH	93.78	94.61	94.70	94.94	89.98	93.17	93.61	67.80
BMSH	93.73	94.51	94.97	95.00	87.97	92.13	93.74	94.81
CorrSH	93.03	94.53	94.28	94.64	87.40	92.81	92.82	93.53
PairRec	94.03	94.67	94.99	95.81	87.38	90.02	93.04	93.64
SNUH	91.54	94.42	94.67	95.13	87.22	91.49	92.91	93.67
DHIM	88.34	94.93	94.31	95.12	81.34	86.51	89.91	85.64
MICPQ	94.87	95.24	94.95	95.68	89.90	93.66	94.14	94.20
HierHash	**95.47**	**95.27**	**95.74**	**95.88**	**90.49**	**93.71**	**95.63**	**95.81**

Table 5: Coarse P@100 on two hierarchical datasets with different numbers of bits in unsupervised document hashing.

$\mathbb{KL}(q_\phi(y_i|s_i)\|p(y_i))$ is further calculated as:

$$\begin{aligned}
&\mathbb{KL}(q_\phi(y_i|s_i)\|p(y_i)) = \mathbb{E}_{q_(y_i|s_i,z_i)}\left[\log\frac{p(y_i)}{q(y_i|s_i)}\right] \\
&= \mathbb{E}_{q(y_i|s_i.z_i)}\left[-\log\frac{e^{\left[z_i^\top\bar{\mu}_{y_i}+\mu_{s_i}^\top\bar{\mu}_{y_i}\right]/\tau_y}}{p(y_i)\sum_{j=1}^{M}e^{\left[z_i^\top\bar{\mu}_j+\mu_{s_i}^\top\bar{\mu}_j\right]/\tau_y}}\right] \\
&\triangleq -\log\frac{e^{\left[z_i^\top\bar{\mu}_{y_i}+\mu_{s_i}^\top\bar{\mu}_{y_i}\right]/\tau_y}}{\sum_{j=1}^{M}e^{\left[z_i^\top\bar{\mu}_j+\mu_{s_i}^\top\bar{\mu}_j\right]/\tau_y}}
\end{aligned} \tag{34}$$

B Complementary Experiment Setups

C Complementary Results

Coarse-grained Experimental Results The coarse-grained experimental results are shown in Table 5 for NYT and DBpedia datasets. It's obvious that HierHash significantly outperforms all baseline methods in all encoding length, demonstrating that the generated hash codes preserve better hierarchical semantics.

Computational Budget The number of parameters in HierHash ranges from 7M to 14M according to the configuration of the number of coarse-grained/fine-grained categories (*i.e.* M, K) and encoding length. During training, our model occupies less than 3G of memories and takes fewer than 30 seconds for an epoch in all datasets of our experiments with an GeForce RTX 2080 GPU.