ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10889846

De-confusing Hard Samples for Text Semantic Hashing

Tian Huang
Shandong University
Jinan, China
huangtian@mail.sdu.edu.cn

Jian Wang†
Shandong University
Jinan, China
wangjian026@126.com

Yuqing Sun*
Shandong University
Jinan, China
sun_yuqing@sdu.edu.cn

***Abstract*—Text semantic hashing maps a text to a compact binary code, which is an important part of information retrieval and language processing. There are two main challenges for this task, one is to make the hash codes express the hierarchical category information for improving the retrieval accuracy, and the other is how to deal with the hard samples. In this paper, we adopt the Bernoulli VAE to encode the text semantics and design the *parent* and *child* level contrastive losses to learn the hierarchical information of the text. To find the hard samples, for each category, we introduce a latent sphere space to split the majority samples and hard samples, where the center and radius are dynamically calculated based on the semantic distance between samples. For the hard samples, we introduce the *de-confusion loss* to pull them close to the center. We conduct experiments on three datasets and the results show that the proposed model outperforms the SOTA baselines. The ablation experiments show that the category constraints and the *de-confusion loss* contribute to the model performance. The results of t-SNE also show that the hash codes learned by our model reflect high category differences.**

***Index Terms*—text hashing, hard sample, hierarchical categories**

I. Introduction

Text semantic hashing aims to map a text to a compact binary code that represents the meaning of the whole text and can be searched by Hamming distance [1], [2]. The hierarchical classification is an important form of organizing documents [3], [4], [5], such as in ACM Computer Classification System, [1] International Patent Classification[2]. These hierarchical category information shows the associations and differences between documents from a macro perspective. Thus it is important to encode the hierarchical category information into the hash codes so that they contain the latent common characteristics of the category [6], [7].

Some methods embed the category information by the additional prediction of text categories [8], [9] or comparison loss [10] in the training phase. These methods only use the flat category information without considering the hierarchical category. To encode the hierarchical category information, the Intra-Category Aware Hierarchical Document Hashing(IHDH) method jointly predicts the probabilities of the child and parent categories [11]. Another method Hierarchical Generative Model (HierHash) introduces hierarchical prototypes to construct a hierarchical prior distribution [12]. These methods do not deal with the hard sample issues, which results in incorrect retrieval results. The hard samples refer to those who have the similar semantics to the samples in different categories, or have the dissimilar semantics to most samples in the same category. For instance, a document that actually belonging to the category "*Atheism*" with many religion-related statements may be close to some documents belonging to the category "*Religion*" in the hash space. Since the categories are independent of specific documents, i.e., there are the dissimilar texts on semantics within one category and similar ones in different categories, ignoring these hard samples will weakens the category differences of hash codes.

In this paper, we propose the **D**e-confusion Hierarchical text **S**emantic **H**ashing model (DSH). Based on the hierarchical category, we adopt the multivariate contrastive loss to bring samples in the same category closer together while separating those in different categories in the hash space. To find the hard samples for a category, we introduce a sphere space to cover the majority samples, the latent center and radius of this sphere space are calculated dynamically by the samples in this category. The samples outside the space are treated as the hard samples. Then a *de-confusion loss* is adopted to constrain the hard samples to be close to their own category center and away from the centers of other categories. The results on three datasets show that the proposed model outperforms all baselines. The ablation experiments show that both the hierarchical category constraints and the *de-confusion loss* improve the model performance.

II. Methodology

A. Problem and Framework

For a text x and its parent category label y^l and the child category label y^p, each child category corresponds to only one parent category, our task aims to learn a function that maps x to a n-dimensional hash code $h \in \{0,1\}^n$. The hash codes should be associated with the category information and reflect the semantic differences between texts. Our proposed method includes three parts, as shown in Fig. 1. The first part learns the text semantics by reconstructing the text. The second

This work was supported by the National Natural Science Foundation of China (62376138) and the Innovative Development Joint Fund Key Projects of Shandong NSF (ZR2022LZH007).

†The author contributed equally to this work.

*Corresponding author

[1]https://dl.acm.org/ccs

[2]http://www.wipo.int/classifications/en/

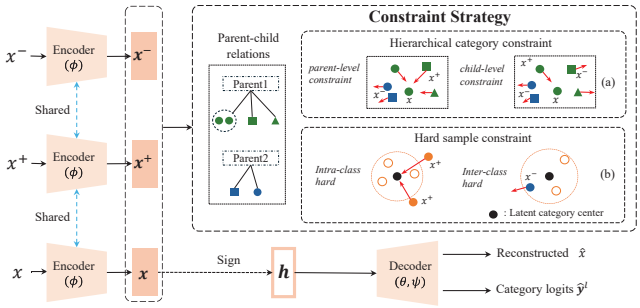

Fig. 1. The DSH model architecture.

part brings samples within the same category closer together and separates them from those in other categories. The third part identifies the hard samples according to the dynamically calculated latent category center and hard radius, then pulls these samples closer to the center.

B. Bernoulli Variational Auto-encoder

We adopt Bernoulli VAE as the encoder-decoder network since the Gaussian distribution can be used to create arbitrary distributions through the rational mapping functions, including the special Bernoulli case for the text hashing [13].

The encoding process, denoted by $f_\phi(h|x)$, first maps an input text x to a continuous vector $\boldsymbol{x}$, each dimension in $\boldsymbol{x}$ is treated as the parameter of a Bernoulli distribution. Then we can sample each dimension of hash code h based on $\boldsymbol{x}$:

$$\boldsymbol{h}_i = \text{sign}(\boldsymbol{x}_i) = \begin{cases} 1 & \text{if } \boldsymbol{x}_i > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The decoding process reconstructs the text from the hash code, denoted by $f_\theta(x|h)$. According to the Bayes' theorem, the above two processes can be optimized simultaneously by maximizing the following loss [14]:

$$\mathcal{L}_{vae}(\theta, \phi) = \mathbb{E}_{f_\phi(h|x)}[\log f_\theta(x|h)] - KL(f_\phi(h|x)||p(h)) \tag{2}$$

where, the Multidimensional Bernoulli distribution $p(h)$ is the assumed distribution of h, i.e., $h \sim \text{Bernoulli}(\rho)$, ρ is the parameter vector of the distribution, KL is the Kullback-Leibler divergence between two probability distributions.

Since the child category provides the base category information, we also use h to predict the child category:

$$\hat{y}_i^l = \frac{\exp(h^T E_y e_i + b_i)}{\sum_{k=1}^{|C_l|} \exp(h^T E_y e_k + b_k)} \tag{3}$$

where E_y is the parameter matrices, and e_i is the one-hot vector corresponding to the i-th category, b_i is the bias term. Let $\hat{y}^l$ denote the predicted child category for x, we construct the category loss as follows:

$$\mathcal{L}_y = \text{CrossEntropy}(\hat{y}^l, y^l) \tag{4}$$

C. Hierarchical Category Constraint

To differentiate the semantically similar samples with different categories and simultaneously aggregate the semantically dissimilar samples within the same category, we introduce the category constraints that include two contrastive losses: the *parent-level* and the *child-level* contrastive loss. We first show the loss of base contrastive learning. Let x, x^+, x^- denote the anchor sample, positive sample, and negative sample, respectively, and $\boldsymbol{x}$ denote the continuous embedding of x, the contrastive loss can be calculated as follows:

$$\mathcal{L}(x, x^+, x^-, m) = \text{relu}\left(\text{d}(\boldsymbol{x}, \boldsymbol{x}^+) - \text{d}(\boldsymbol{x}, \boldsymbol{x}^-) + m\right) \tag{5}$$

where, m is the margin hyper-parameter, $\text{d}(\cdot, \cdot)$ is the distance metric. To enable the gradient calculation, we use Euclidean distance as $\text{d}(\cdot, \cdot)$ rather than the Hamming distance.

For the *parent-level* contrastive loss $\mathcal{L}_t^p$, the positive sample set X_p^+ contains the samples that have different child categories but the same parent category with x, and the negative sample set X_p^- contains the samples that have different parent categories with x. Specially, the $\mathcal{L}_t^p$ is defined as follows:

$$\mathcal{L}_t^p = \mathbb{E}_{x^+ \in X_p^+, x^- \in X_p^-} \mathcal{L}(x, x^+, x^-, m_p) \tag{6}$$

For the *child-level* contrastive loss $\mathcal{L}_t^l$, the positive sample set X_l^+ contains the samples that are in the same child category as x, X_l^- denotes the samples that have different child categories with x. The formulation of $\mathcal{L}_t^l$ is:

$$\mathcal{L}_t^l = \mathbb{E}_{x^+ \in X_l^+, x^- \in X_l^-} \mathcal{L}(x, x^+, x^-, m_l) \tag{7}$$

The entire *hierarchical category loss* is transformed into an objective function by the hyper-parameters α:

$$\mathcal{L}_h = \alpha \mathcal{L}_t^p + (1 - \alpha)\mathcal{L}_t^l \tag{8}$$

D. Hard Sample Constraint

The hard samples seriously affect the learning of hash space, to deal with this, for each category, we compute a sphere space for finding the hard samples, where the sphere space is restricted by the latent center and radius, samples outside the space are the hard samples. Then we constrain these samples by a *de-confusion* loss,

Let C denote the set of samples that have the same category with x. The center of category C, denoted as $\boldsymbol{\mathcal{C}}$, is located at the average embeddings for the samples in C. We assume the distance from samples to a category center follow a Gaussian distribution $\mathcal{N}(\mu, \sigma)$. The mean μ is calculated as the expectation of the distance from all samples in this category to the center: $\mu = \mathbb{E}_{x \in C} \text{d}(\boldsymbol{x}, \boldsymbol{\mathcal{C}})$. The variance σ is calculated by $\sigma = sqrt\left(\mathbb{E}_{x \in C}(\text{d}(\boldsymbol{x}, \boldsymbol{\mathcal{C}}) - \mu)^2\right)$. Then, a hard radius $\mathcal{R} = \mu + \lambda\sigma$ is computed to pick up the hard samples, where λ is a hyper-parameter.

For a category C and its hard radius $\mathcal{R}$, the hard samples can be divided into two cases, named *intra-class hard samples* and *inter-class hard samples*. The former refers to the samples that belong to C while their distance to the category center is larger than $\mathcal{R}$. The latter refers to the samples that do not belong to category C while the distance to the category center

TABLE I
DATASET STATISTICS

Datasets	Domain	Train	Test	Parent	Child	Feature
20News	News	11314	7532	7	20	2000
RCV1	News	39832	26556	3	29	31812
WOS	Science	28089	18726	7	134	10000

is less than $\mathcal{R}$. We introduce the *de-confusion loss* $\mathcal{L}_d$ to pull hard samples close to their category centers while moving away from from other centers. Let X_c^{l+}, X_c^{l-} denote the sample set selected from the *intra-class hard samples* and the *inter-class hard samples* based on the child category, respectively. X_c^{p+} and X_c^{p-} denote the sample set selected according to the parent category, respectively. Then, $\mathcal{L}_d$ can be calculated as following:

$$\mathcal{L}_d = \alpha\mathcal{L}_c^p + (1-\alpha)\mathcal{L}_c^l, \tag{9}$$

$$\mathcal{L}_c^p = \mathbb{E}_{x^+\in X_c^{p+}, x^-\in X_c^{p-}}\, \mathcal{L}(\boldsymbol{\mathcal{C}}_p, x^+, x^-, m_p^c), \tag{10}$$

$$\mathcal{L}_c^l = \mathbb{E}_{x^+\in X_c^{l+}, x^-\in X_c^{l-}}\, \mathcal{L}(\boldsymbol{\mathcal{C}}_l, x^+, x^-, m_l^c) \tag{11}$$

where, $\mathcal{L}_c^p$ and $\mathcal{L}_c^l$ are the *de-confusion loss* for the parent and child categories with hyper-parameters m_p^c and m_l^c, respectively. $\boldsymbol{\mathcal{C}}_p$ and $\boldsymbol{\mathcal{C}}_l$ refer to the parent and child category centers, respectively.

E. Overall Objective

We optimize our DSH model by combining multiple objective functions, where β, γ, and ξ are hyper-parameters representing the weights of each part:

$$\mathcal{L}_{DSH} = \mathcal{L}_{vae} + \beta\mathcal{L}_y + \gamma\mathcal{L}_h + \xi\mathcal{L}_d, \tag{12}$$

III. EXPERIMENTS

A. Datasets and Evaluation Metrics

We use the hash codes for information retrieval and conduct experiments on there benchmark datasets: 20Newsgroups (20News)[3], Reuters Corpus Volume I (RCV1)[4], Web of Science (WOS)[5]. Table I shows the details of datasets. We adopt the widely used Precision@K and NDCG@K to evaluate the performance of our model [15], [16], [11]. Precision@K is the proportion of retrieved top K samples that have the same child category with query. NDCG@K takes into account both the relevance of the items and their positions in the ranking, giving higher weight to results that appear earlier. As a regular way, we use the hash codes of the test samples as queries and search K similar samples in the training set by Hamming distance.

B. Comparison Methods and Training Details

We compare DSH with multiple baselines. SHTTM [6] adopts tags and topic model for semantic hashing. VDSH-S and VDSH-SP [8] use VAE to learn the hash codes. NASH-DN-S [9] is a neural architecture for semantic hashing with data-dependent noise decoding. GMSH-S, BMSH-S [15] and PSH [10] are mixture-Prior based hashing methods. IHDH [11] is the SOTA method with hierarchical category information.

[3]http://qwone.com/~jason/20Newsgroups/
[4]https://data.mendeley.com/datasets/9rw3vkcfy4/6
[5]https://scikit-learn.org/0.18/datasets/rcv1.html

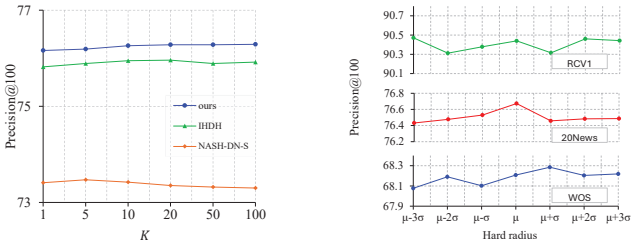

Fig. 2. Analysis of hard radius and K on Precision@100.

In our network, the MLP has 1000 neurons and leaky ReLU activation units. To prevent over-fitting, we set a dropout probability of 0.1 after the second layer [17]. We compute the category centers and hard radius every 3 epochs. For the prior Bernoulli distribution $p(h) = \text{Bernoulli}(\rho)$, we set $\rho = \mathbf{0.5}$, which can maximizes the information entropy of hash codes. For the non-differentiable phenomenon of sample truncation, we use the straight-through(ST) estimator[18] for gradient estimation. We set a learning rate of 0.0005 and the maximum number of epochs to 50, $\lambda = 1, \alpha = 0.55, \beta = 3, \gamma = 3, \xi = 1$. m_h^p and m_c^p are set to 0.01, m_h^l and m_c^l are set to 0.1.

C. Results

The results in Tab II show that DSH outperforms the comparison methods including the SOTA method IHDH on all hashing bits and datasets. What's more, on the difficult dataset WOS which contains more parent and child categories than other datasets, DSH achieves the largest improvement over the existing methods. To illustrate the stability of DSH, we analyze the Precision@K on 20News under 64 bits with different values of K. The results in Fig. 2 show that our model outperforms the compared models at all listed K values.

To compare the different ways of applying hierarchical information, we compare the results of IHDH-HC with DSH-HC, where IHDH-HC is only trained by jointly predicting the probabilities of the child and parent categories. DSH-HC is our method trained only using the *hierarchical loss*. The results in Table III show that DSH-HC outperforms the IHDH-HC by an average of 0.61 on Precision@100 and 0.65 on NDCG@100. This shows the advantage of our *hierarchical loss*.

D. Ablation Study and Hyperparameters Analysis

We conduct ablation experiments to show the effects of two key losses: (1) w/o $\mathcal{L}_h$ refers to removing *hierarchical loss*. (2) w/o $\mathcal{L}_d$ refers to removing the *de-confusion loss*. Results in Table IV show that each loss plays an important role, especially the *hierarchical loss*. When removing the *hierarchical loss*, the average Precision@100(denoted as P) and NDCG@100(denoted as N) of DSH are reduced by 2.9 and 3.0, respectively.

For the key hyperparameters: λ controls the hard radius $\mathcal{R}$. α controls the contributions of parent constraint and child constraint in *hierarchical loss*. β, γ, ξ are used to balance the loss in the overall objective. Results in Fig. 2 show that for most datasets, a moderate λ such as 0 or 1 is reasonable, since

TABLE II
COMPARISON RESULTS OF PRECISION@100 AND NDCG@100

Methods	WOS				RCV1				20News			
	16bits	32bits	64bits	128bits	16bits	32bits	64bits	128bits	16bits	32bits	64bits	128bits
SHTTM	20.13/20.01	11.7/11.53	7.54/7.47	6.32/6.12	74.22/73.65	62.13/61.53	57.34/56.94	55.87/55.17	35.36/35.32	34.23/34.19	25.75/25.73	23.11/23.10
VDSH-S	36.44/36.08	50.95/50.57	51.12/50.79	51.85/51.55	86.11/83.51	88.12/85.62	88.85/86.47	89.44/87.13	66.10/65.92	70.89/70.81	70.29/70.26	71.87/71.84
VDSH-SP	37.45/37.18	52.31/52.17	51.44/51.99	52.73/52.98	85.21/82.33	87.62/84.72	88.25/85.31	89.14/86.03	66.30/64.32	69.20/69.73	70.39/70.28	70.77/70.85
NASH-DN-S	5.90/5.74	10.35/10.22	21.24/20.77	34.71/34.24	58.55/55.88	76.19/73.25	87.81/84.86	88.13/85.47	59.47/59.61	73.98/73.88	73.78/73.77	67.18/67.08
GMSH-S	44.23/44.22	53.55/53.31	56.20/55.21	53.56/53.09	85.11/83.12	86.42/84.12	85.34/83.62	84.27/82.34	69.99/69.81	70.32/70.08	69.73/69.69	68.21/68.11
BMSH-S	49.27/49.05	56.64/56.52	57.40/57.2	55.41/55.04	87.39/85.06	88.56/86.18	87.89/85.52	86.60/84.12	71.87/71.92	72.18/72.15	71.83/71.18	70.20/70.01
PSH-ARM	26.17/25.52	46.55/45.88	56.62/56.24	62.99/62.86	86.90/84.20	88.70/86.12	89.20/86.53	89.99/87.66	73.17/73.11	74.81/74.79	75.24/75.24	75.56/75.56
IHDH	50.00/49.64	58.51/58.25	62.42/62.27	63.77/63.67	87.44/85.07	88.74/86.66	89.24/86.63	90.12/87.74	73.60/73.59	75.58/75.59	76.14/76.14	76.42/76.41
ours	**54.24/54.19**	**61.40/61.40**	**66.41/66.46**	**68.21/68.29**	**87.96/85.24**	**89.54/86.76**	**90.16/87.34**	**90.44/87.75**	**74.33/74.32**	**75.91/75.90**	**76.29/76.29**	**76.67/76.67**

The results of baselines are from paper [11].

TABLE III
COMPARISON RESULTS ON DIFFERENT HIERARCHICAL SCHEMES.

Metrics & Methods		16bits	32bits	64bits	128bits
Precision@100	IHDH-HC	72.49	74.82	75.13	75.68
	DSH-HC	**73.65**	**75.25**	**75.75**	**75.92**
NDCG@100	IHDH-HC	72.42	74.83	75.13	75.57
	DSH-HC	**73.63**	**75.25**	**75.75**	**75.93**

TABLE IV
ABLATION STUDY.

Model	20News		RCV1		WOS	
	P	N	P	N	P	N
DSH	**76.67**	**76.67**	**90.44**	**87.75**	**68.21**	**68.29**
w/o $\mathcal{L}_h$	71.91	71.88	89.08	86.29	65.56	65.54
w/o $\mathcal{L}_d$	75.92	75.93	90.21	87.38	68.02	68.10

a large λ degrades the text reconstruction ability of the hashing codes, while a small value may not separate the hard samples within categories. Results in Fig. 3 also show that the model is robust to parameters α, β, ξ over a large interval, such as the range of α in [0.1, 0.8]. The DSH model is sensitive to γ within interval [3, 10] since the strong hierarchical constraints may break the reconstruction goal in Equation 2.

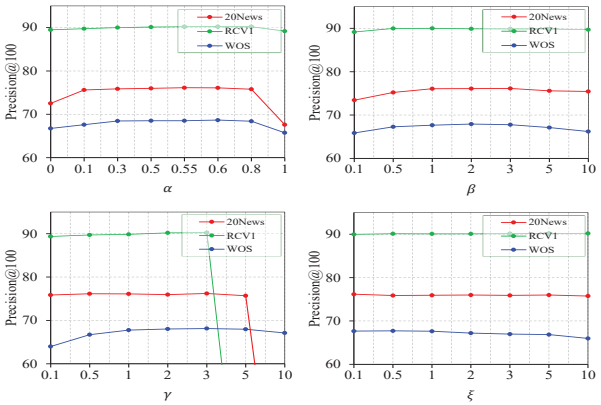

Fig. 3. Loss weights analysis on Precision@100.

For the margin hyper-parameters, Fig. 4 shows that DSH is robust to low m_p. Appropriate values for m_l can maintain the clustering trend of parent categories while separating samples within different child categories in hash space. High m_p^c and m_l^c affect the text reconstruction and lead to low performance.

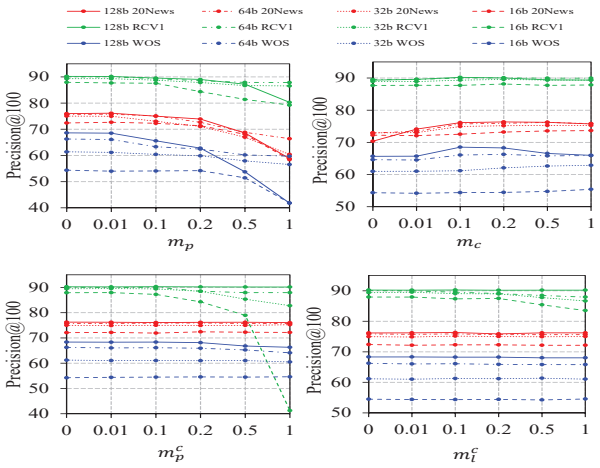

Fig. 4. Analysis of margins on Precision@100.

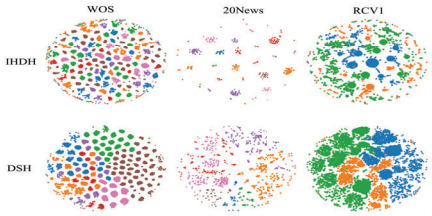

Fig. 5. Visualization of the parent categories on three datasets.

E. Hash Space Visualization

We employ t-SNE [19] to visualize the distribution of hash codes learned by DSH and IHDH. The results in Fig. 5 show that the samples within same parent category (indicated by the same color) are closer than those in different ones, while the results of IHDH do not have this trend. This shows the effectiveness of DSH in utilizing the hierarchical information.

IV. CONCLUSION

In this paper, we propose the Semantic Hashing model DSH. We use the hierarchical category constraint to encode both the text semantics and hierarchical category information into the hash codes. Additionally, we introduce the latent center and hard radius to represent the latent common characteristics of each category. Then we identify the hard samples. We introduce the *de-confusion loss* to draw back these hard samples to the center. Experiments on three benchmark datasets show that DSH outperforms all baselines.

REFERENCES

[1] Ruslan Salakhutdinov and Geoffrey Hinton, "Semantic hashing," *International Journal of Approximate Reasoning*, vol. 50, no. 7, pp. 969–978, 2009.

[2] Liyang He, Zhenya Huang, Enhong Chen, Qi Liu, Shiwei Tong, Hao Wang, Defu Lian, and Shijin Wang, "An efficient and robust semantic hashing framework for similar text search," *ACM Trans. Inf. Syst.*, vol. 41, no. 4, Mar. 2023.

[3] Aixin Sun and Ee-Peng Lim, "Hierarchical text classification and evaluation," in *Proceedings 2001 IEEE International Conference on Data Mining*. IEEE, 2001, pp. 521–528.

[4] Daphne Koller and Mehran Sahami, "Hierarchically classifying documents using very few words," in *ICML*, 1997, vol. 97, pp. 170–178.

[5] Juho Rousu, Craig Saunders, Sandor Szedmak, and John Shawe-Taylor, "Learning hierarchical multi-category text classification models," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 744–751.

[6] Qifan Wang, Dan Zhang, and Luo Si, "Semantic hashing using tags and topic modeling," in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, 2013, pp. 213–222.

[7] Wei Dong, Qinliang Su, Dinghan Shen, and Changyou Chen, "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)*, 2019, pp. 5226–5235.

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

[9] Dinghan Shen, Qinliang Su, Paidamoyo Chapfuwa, Wenlin Wang, Guoyin Wang, Lawrence Carin, and Ricardo Henao, "Nash: Toward end-to-end neural architecture for generative semantic hashing," *arXiv preprint arXiv:1805.05361*, 2018.

[10] Siamak Zamani Dadaneh, Shahin Boluki, Mingzhang Yin, Mingyuan Zhou, and Xiaoning Qian, "Pairwise supervised hashing with bernoulli variational auto-encoder and self-control gradient estimator," in *Conference on Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 540–549.

[11] Jia-Nan Guo, Xian-Ling Mao, Wei Wei, and Heyan Huang, "Intra-category aware hierarchical supervised document hashing," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 6003–6013, 2023.

[12] Qian Zhang, Qinliang Su, Jiayang Chen, and Zhenpeng Song, "Document hashing with multi-grained prototype-induced hierarchical generative model," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, Eds., Miami, Florida, USA, Nov. 2024, pp. 321–333, Association for Computational Linguistics.

[13] Carl Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[14] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *stat*, vol. 1050, pp. 1, 2014.

[15] Casper Hansen, Christian Hansen, Jakob Grue Simonsen, Stephen Alstrup, and Christina Lioma, "Unsupervised neural generative semantic hashing," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 735–744.

[16] Suthee Chaidaroon, Dae Hoon Park, Yi Chang, and Yi Fang, "node2hash: Graph aware deep semantic text hashing," *Information Processing & Management*, vol. 57, no. 6, pp. 102143, 2020.

[17] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, pp. 1929–1958, 2014.

[18] Yoshua Bengio, Nicholas Léonard, and Aaron Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[19] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.