SchemaTopic: Neural Topic Modeling with LLM-Generated Schemas

Anonymous ACL submission

Abstract

Topic modeling aims at discovering coherent and interpretable latent themes from text corpora. Although recent LLM-based topic modeling methods show strong semantic capabilities, they still face limited structural organization, reliance on a predefined number of topics, and high cost. To address these issues, we propose SCHEMATOPIC, a framework that injects schema-guided LLM refinement into neural topic models (NTMs). SCHEMATOPIC first learns an initial latent topic space with an NTM and then uses an LLM to induce topic schemas by identifying partitioning criterion, pruning low-quality topics, and assigning schema labels to the remaining topics. The refined outputs are used as anchor signals to retrain the NTM with an anchor-guided loss. By combining the strengths of NTMs and LLMs, SCHEMATOPIC improves topic quality and interpretability while reducing cost by limiting LLM usage to schema induction. Extensive experiments show that SCHEMATOPIC produces more coherent and structured topics with enhanced interpretability under substantially reduced LLM overhead.

1 Introduction

Topic modeling is a fundamental task in natural language processing that uncovers latent thematic structure from document collections in an unsupervised manner. Classical probabilistic models such as Latent Dirichlet Allocation(LDA) have long provided interpretable corpus-level summaries (Blei et al., 2003), and more recent neural topic models(NTMs) have further improved flexibility in modeling document-topic and topic-word distributions (Miao et al., 2017; Srivastava and Sutton, 2017a; Dieng et al., 2020a; Wu et al., 2023a, 2024a). Recent studies have also begun to leverage large language models (LLMs) to improve topic quality and interpretability, reflecting the growing interest in combining corpus-level topic modeling with the semantic strengths of LLMs (Pham et al., 2024; Yang et al., 2025).

Despite this progress, existing approaches still face three main limitations. First, many topic models represent topics as flat top-word lists, leaving relations among topics under-modeled and requiring manual interpretation (Wu et al., 2024b). Second, even when a hierarchical structure is considered, the number of topics K is often fixed in advance, while determining an appropriate topic number remains a longstanding challenge in topic modeling (Bulatov et al., 2024; Schroeder et al., 2025). Third, recent LLM-based topic modeling methods are often costly or hard to scale because they rely on repeated LLM calls (Yang et al., 2025; Pham et al., 2024). Even efficient LLM-assisted methods still have room to improve topic reorganization and adapt the effective number of topics (Yang et al., 2025).

To address these limitations, we propose **SCHEMATOPIC**, a schema-guided framewor$K$ for structured neural topic modeling. SCHEMATOPIC first trains an NTM to obtain an initial latent topic space. It then uses an LLM to induce a topic schema—consisting of a semantic partitioning criterion and its associated categories—by identifying the criterion, pruning low-quality topics, and assigning the remaining topics to the induced categories. The resulting schema structure is flattened into topic-level anchor word sets and used to retrain the NTM with an anchor-guided loss. This process improves topic organization and interpretability while preserving the modeling strengths of NTMs.

Our contributions are summarized as follows:

- **Schema-guided topic refinement**: We introduce an LLM-based refinement pipeline that induces schemas, prunes low-quality topics, and assigns topics to induced categories, producing a more structured and interpretable topic space.

- **Automatic determination of K**: Rather than specifying the topic number K, SCHEMATOPIC starts from an initial K and automatically determines the effective number of topics through schema-guided pruning and reassignment.

- **Cost efficiency**: SCHEMATOPIC enables cost-efficient LLM integration by operating at topic refinement, avoiding the extensive and repeated queries common in prior LLM-based topic modeling methods.

2 Background

2.1 Topic Modeling

Topic modeling aims to uncover latent semantic themes in a corpus by representing each document as a mixture of topics and each topic as a distribution over words (Blei et al., 2003).

Let $\mathcal{D} = \{d_1, d_2, \ldots, d_M\}$ denote a corpus of M documents, where each document d is represented as a sequence or multiset of words drawn from a vocabulary $\mathcal{V}$ of size V. Topic modeling learns a set of K latent topics $\mathcal{K} = \{1, \ldots, K\}$, where each topic $\beta_k \in \Delta^{V-1}$ is defined as a probability distribution over the vocabulary, and each document is associated with a topic proportion $\theta_d \in \Delta^{K-1}$.

Under this formulation, a document can be interpreted as being generated by a mixture of topics, where each topic contributes words according to its corresponding distribution over the vocabulary. The objective of topic modeling is to estimate:

$$\{\beta_k\}_{k=1}^{K} \quad and \quad \{\theta_d\}_{d=1}^{M}, \tag{1}$$

such that the observed corpus $\mathcal{D}$ is well explained by these latent variables.

2.2 Neural Topic Modeling Framework

Neural topic models (NTMs) extend classical topic modeling by parameterizing topic and document representations using neural networks (Miao et al., 2017; Srivastava and Sutton, 2017a).

Given a document represented as a bag-of-words vector $\mathbf{x} \in \mathbb{R}^V$, an NTM learns K latent topics, represented by the topic-word matrix $\beta \in \mathbb{R}^{V \times K}$ and infers a document-level topic distribution $\theta \in \Delta^K$.

Most NTMs follow a variational autoencoder (VAE) framework. The encoder maps $\mathbf{x}$ to a latent variable $\mathbf{z}$ with approximate posterior

$$q_\Theta(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}(\mu, \Sigma), \tag{2}$$

from which the document-topic distribution is obtained as $\theta = \mathrm{softmax}(\mathbf{z})$. The decoder reconstructs the input document via $p_\Phi(\mathbf{x} \mid \theta)$.

Training is based on the evidence lower bound (ELBO) (Kingma and Welling, 2014):

$$\mathcal{L} = -\mathbb{E}_{q_\Theta}\left[\log p_\Phi(\mathbf{x} \mid \theta)\right] + \mathrm{KL}\left(q_\Theta \,\|\, p(\mathbf{z})\right). \tag{3}$$

3 Related Work

Neural Topic Models. Neural topic models (NTMs) improve the flexibility of classical topic models by learning document-topic and topic-word distributions with deep latent-variable architectures (Miao et al., 2017; Srivastava and Sutton, 2017a). Among them, ETM (Dieng et al., 2020a) and ECRTM (Wu et al., 2023a) improve topic quality by incorporating semantic embedding structure into neural topic modeling. Prior work has also explored structured topic modeling beyond flat topic sets, including models that capture hierarchical or relational organization among topics (Miyamoto and Sasano, 2023). In addition, clustering-based approaches such as BERTopic allow the effective number of topics to emerge from the data rather than being fixed in advance (Grootendorst, 2022). However, many existing methods still either rely on a predefined topic number or do not explicitly enforce semantically interpretable and structurally organized topic representations.

LLMs in Topic Modeling. Recent studies have incorporated LLMs into topic modeling in several ways. Some methods use LLMs to directly generate or organize topics from document collections, as in TopicGPT (Pham et al., 2024). Other approaches combine LLMs with neural topic models to refine topic quality while preserving the efficiency of NTM-based document modeling, as in LLM-ITL (Yang et al., 2025). LLMs have also been used for post-hoc topic refinement by revising low-quality topic words (Chang et al., 2024), and more recent work explores cost-aware LLM-assisted topic augmentation with limited LLM calls (Chang et al., 2025). Although these approaches improve semantic quality and interpretability, they still do not explicitly induce a schema that restructures topics and adapts the effective topic set. In contrast, our method leverages LLM-based schema reasoning to refine NTM-derived topics, enabling semantics-driven pruning and structured organization with low LLM cost.

4 Method

In this work, we propose SCHEMATOPIC, a two-phase framework that combines neural topic modeling with LLM-based schema induction. Figure 1 illustrates the overall pipeline of SCHEMATOPIC. First, an NTM is trained to obtain an initial topic-word space. Then, an LLM refines these topics by inducing schemas, pruning low-quality topics, and assigning the remaining topics to schema labels. Based on the refined outputs, SCHEMATOPIC constructs anchor signals and retrains the NTM with an anchor-guided objective. This process improves topic interpretability while inducing schema-level and topic-level structure.

4.1 Warm-up Training

Given a corpus with vocabulary size V, we first train a standard neural topic model to learn K latent topics, obtaining a topic–word distribution matrix $\beta \in \mathbb{R}^{K \times V}$. These topics provide a corpus-grounded semantic space, but they may still contain noisy, redundant, or weakly interpretable topics. We extract the top-W words for each topic (i.e., the W highest-probability words in each row of β; W=15 in all experiments) and use them as structured inputs to the subsequent LLM-based refinement stage.

4.2 LLM-based Topic Refinement

The LLM refinement phase consists of three steps: schema design, topic pruning, and schema assignment. Each step is realized by prompting the LLM with the full topic set and the schema context. First, the LLM induces a set of high-level schemas from the initial topics, providing a semantic partitioning criterion over the topic space. It then evaluates each topic and removes topics that are redundant, vague, or poorly aligned with the induced schemas. Finally, it assigns each remaining topic to a schema and produces schema-aligned topic descriptors, yielding a refined and structured topic set.

We provide two operational modes for the refinement pipeline. In **auto mode**, topics judged as low-quality in Step 2 are permanently removed, and the effective topic count K' may be smaller than the initial K. In **keep mode** (K-keep), deleted topics are not discarded but reassigned to a residual "Misc" schema category and passed to Step 3 alongside the surviving topics, so that the final topic count always equals K.

4.2.1 Schema Design

Given the initial topic-word distributions produced by the warm-up NTM, we prompt the LLM with all K top-word lists and instruct it to identify a global partitioning criterion over the full topic set and generate a single flat schema set. Rather than labeling topics independently, the LLM is asked to reason about how the entire topic space can be organized under a shared semantic principle. The LLM outputs a structured response consisting of the criterion (one sentence) and a flat list of broad category labels (typically 5–20). Any inadvertent hierarchical structure in the output is flattened and duplicate labels are removed. The resulting schema labels $\mathcal{S} = \{s_1, \ldots, s_M\}$ serve as global decision references in the following steps, where they are used to guide topic pruning and schema assignment.

4.2.2 Topic Pruning

Using the induced schema set as context, the LLM then evaluates each topic and makes a binary quality decision (*keep* or *weak*). The prompt explicitly instructs the model to assess whether a topic is interpretable, whether it is compatible with the schema structure, and whether it represents a specific semantic theme rather than a generic, vague, mixed, or noisy word set. For retained topics, the LLM also produces a short topic name (1–2 words). Topics are marked as weak only when they are clearly unclear, overly generic, semantically mixed, dominated by stopwords, or cannot be assigned a reasonable name. The LLM is required to output exactly K decision objects, one per topic. This step filters out low-quality topic candidates while preserving semantically meaningful ones for the next step.

In auto mode, weak topics are permanently removed and only the surviving topics proceed to Step 3, re-indexed to consecutive indices $\{0, 1, \ldots\}$. In keep mode, weak topics are instead relabeled with placeholder names (`misc1`, `misc2`, ...) and assigned to the Misc schema category; both surviving and Misc topics are then passed to Step 3.

4.2.3 Schema Assignment

For the surviving topics (and, in keep mode, the Misc topics as well), we perform a schema-aware refinement step in which the LLM jointly updates topic words and assigns each topic to the most appropriate schema label. The model is instructed

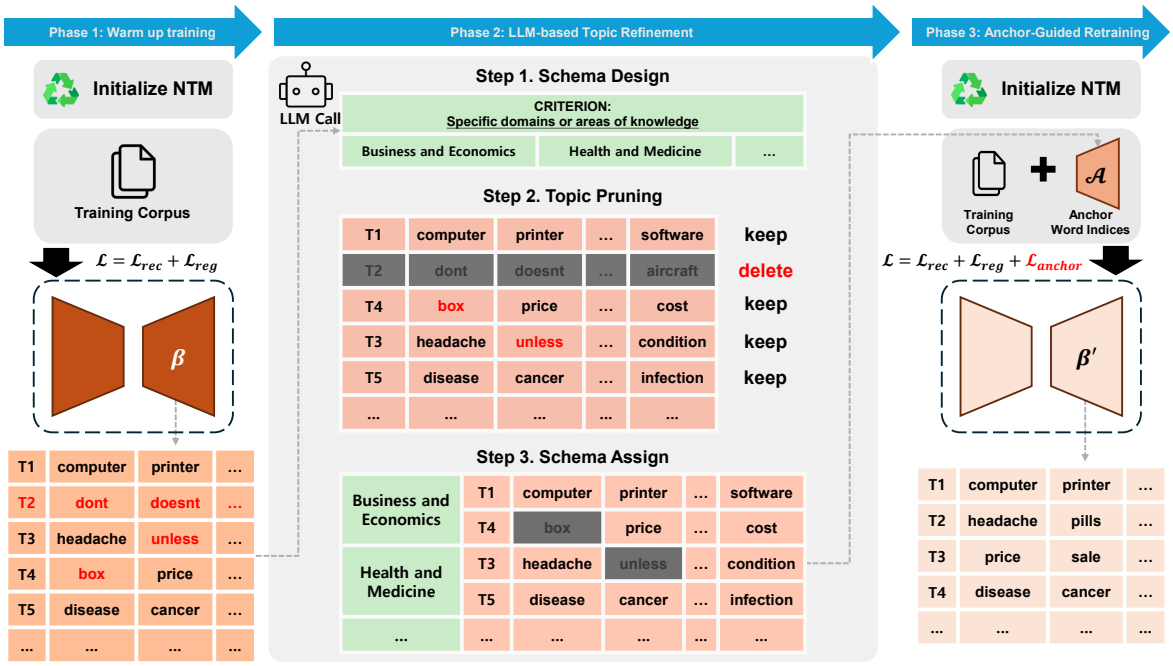

Figure 1: Overview of the SchemaTopic framework. The proposed method consists of three phases. (Phase 1) A neural topic model (NTM) is first trained on the corpus to obtain initial topic–word distributions. (Phase 2) An LLM is called to improve topic quality through (a) schema design, (b) topic pruning, and (c) schema assignment. (Phase 3) An anchor topic-word distribution is generated from the refined schema–topic structure, which is used to retrain the NTM from scratch with an anchor-guided objective.

to preserve a tight semantic cluster, remove only words that are clearly generic or inconsistent with the topic, and retain 6–8 representative words per topic. At this step, a second round of pruning is also performed: if a topic remains too weak or unclear even after refinement, or if it cannot be plausibly assigned to any schema, it is deleted (auto mode) or assigned to Misc (keep mode). The final output is therefore a structured topic set in which each retained topic is associated with a refined lexical representation and a schema label.

After the LLM call, rule-based post-processing is applied: tokens with fewer than 3 characters are removed, standard English stopwords are filtered, and words appearing in two or more topics are removed to reduce lexical overlap. In keep mode, any topic ID missing from the LLM output is recovered from the Step 2 result with schema label Misc, guaranteeing that the final topic count equals K.

4.3 Anchor-Guided Retraining

After obtaining the refined schema-aligned topics, we convert the hierarchical schema–topic–word structure into topic-level anchor word sets for retraining. Although the LLM refinement stage produces a structured hierarchy in which schemas contain multiple topics and each topic is associated with a refined word set, the underlying neural topic model remains flat and does not explicitly model hierarchical relations. We therefore flatten the refined hierarchy at the topic level and represent each retained topic k by an anchor word index set $\mathcal{A}_k \subseteq \{1, \ldots, V\}$, constructed by mapping the refined words back to vocabulary indices; words absent from the corpus vocabulary are discarded. These topic-specific anchor sets are then used to retrain the neural topic model from scratch, with $K' = |\mathcal{T}_{\text{final}}|$ topics in auto mode or $K' = K$ topics in keep mode.

Let $\beta \in \mathbb{R}^{K' \times V}$ denote the topic-word distribution, where K' is the number of topics and V is the vocabulary size. For each topic k, $\mathcal{A}_k$ denotes the set of anchor word indices obtained from the flattened refined topic representation. We define the anchor regularization term as

$$\mathcal{L}_{\text{anchor}} = \lambda \cdot \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \left(-\frac{1}{|\mathcal{A}_k|} \sum_{v \in \mathcal{A}_k} \log \beta_{k,v} \right), \tag{4}$$

where $\mathcal{K}$ is the set of topics with at least one anchor word and λ is the anchor regularization weight. This objective encourages the model to assign higher probability mass to anchor words within each topic. Topics with $\mathcal{A}_k = \emptyset$ do not contribute to $\mathcal{L}_{\text{anchor}}$.

During retraining, the anchor loss is added to

the model's original regularization terms and optimized jointly with the reconstruction objective. The final training loss is defined as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{anchor}}, \quad (5)$$

where $\mathcal{L}_{\text{rec}}$ denotes the reconstruction loss and $\mathcal{L}_{\text{reg}}$ includes the default regularization terms of the underlying neural topic model, such as KL-based penalties. In this way, retraining preserves the corpus modeling ability of the original NTM while aligning the learned topic-word distributions with the refined schema structure.

5 Experiments

5.1 Experimental Settings

Datasets. We evaluate our method on four benchmark corpora: 20News (Lang, 1995), AGNews (Zhang et al., 2015), R8 (Lewis et al., 2004), and DBpedia (Lehmann et al., 2015). These datasets cover diverse domains and label granularities, providing a broad test bed for topic quality and structure. We apply a unified preprocessing pipeline across all datasets and summarize the statistics in Table

Baselines. We compare against five neural topic modeling baselines: ETM (Dieng et al., 2020b), ECRTM (Wu et al., 2023b), NSTM (Zhao et al., 2021), NVDM (Miao et al., 2016), and ProdLDA (Srivastava and Sutton, 2017b). Our method augments these NTM backbones with large language models for schema induction and refinement. We use LLaMA-8B (Touvron et al., 2023) as the default model, and additionally experiment with GPT-4 (OpenAI, 2023) and Claude Opus (Anthropic, 2024) to analyze robustness across different LLMs.

Evaluation Metrics. We evaluate topic quality using C_V coherence (Röder et al., 2015) and topic diversity (TD), and further assess structural consistency using the Silhouette Score (Rousseeuw, 1987). C_V measures the semantic coherence of top words within each topic, while TD evaluates lexical diversity across topics. The Silhouette Score captures how well topics are separated in the embedding space, reflecting the structural quality of the learned schema–topic organization.

Implementation Details. For each dataset, we train the NTM for 250 epochs during both the warm-up phase and the anchor-guided retraining phase. We use LLaMA 8B as the default LLM for the refinement pipeline. After the warm-up training, we apply the LLM-based refinement, followed by retraining the model from scratch with the anchor-guided objective. For both training phases, we use a learning rate of 0.001 and adopt the Adam optimizer (Kingma and Ba, 2015). Unless otherwise specified, the anchor weight λ_{anchor} is set to 1.0, and 2.0 for the R8 dataset. All experiments are conducted on a server equipped with a single NVIDIA RTX A6000 GPU (48GB VRAM).

6 Results

6.1 Topic Quality

To evaluate the overall topic quality, we analyze semantic and structural metrics across datasets as summarized in Table 1. As introduced in the previous section, our framework enables the automatic determination of the effective number of topics K through LLM-based refinement. We evaluate our method under two modes: **keep** mode and **auto** mode. In keep mode, the number of topics K is fixed throughout refinement, while auto mode allows the LLM to iteratively prune topics through a three-step process, automatically determining the effective K. Unless otherwise specified, we initialize with $K = 50$ (see Appendix for other settings). We use Llama 8B as the default setting, and the performance of GPT-5.2 is reported in Appendix Table 6.

Auto mode yields the largest gains. A key observation is that auto mode consistently achieves comparable or better performance than keep mode, despite not relying on a fixed topic count. For example, in ETM on AGNews, auto mode improves C_V from 0.4748 to 0.4997 and significantly boosts the Silhouette score from 0.1443 to 0.3693. Similar trends are observed across datasets, indicating that LLM-guided pruning effectively removes redundant or low-quality topics while improving both semantic and structural quality.

Stronger structural quality. Compared with LLM-ITL (Yang et al., 2025), our method shows clear advantages in structural quality. In particular, Silhouette scores are consistently higher across all datasets and models, with improvements of up to +0.45. These gains indicate that schema-guided refinement produces better-separated and more well-formed topic clusters in the embedding space.

Model	20News			AGNews			DBPedia			R8		
	C_V	TD	Silhouette	C_V	TD	Silhouette	C_V	TD	Silhouette	C_V	TD	Silhouette
ETM	0.4799	0.7016	-0.0482	0.4620	0.7832	0.1002	0.4863	0.7968	0.2777	0.3936	0.6096	0.1124
w. LLM ITL	0.4862	0.6976	-0.0456	0.5178	0.7808	0.0478	0.5363	0.8120	0.2249	0.4510	0.5912	0.1736
Δ	0.0063	-0.0040	0.0026	0.0558	-0.0024	-0.0524	0.0500	0.0152	-0.0528	0.0574	-0.0184	0.0612
w. schema K	0.4803	0.7288	-0.0023	0.4748	0.8616	0.1443	0.4945	0.8360	0.2251	0.4154	0.6736	0.1117
Δ	0.0004	0.0272	0.0459	0.0127	0.0784	0.0441	0.0081	0.0392	-0.0525	0.0218	0.0640	-0.0007
w. schema A	0.4865	0.8314	0.1801	0.4997	0.8923	0.3693	0.4863	0.9015	0.5145	0.3995	0.7376	0.1242
Δ	0.0066	0.1298	0.2283	0.0377	0.1091	0.2691	0.0000	0.1047	0.2369	0.0060	0.1280	0.0118
ECRTM	0.4460	0.7728	0.0000	0.3856	1.0000	0.0000	0.4794	1.0000	0.0000	0.3561	0.8712	0.0000
w. schema K	0.4174	0.7976	0.0595	0.3873	1.0000	0.0595	0.4663	0.9976	0.3649	0.3330	0.8944	0.1832
Δ	-0.0286	0.0248	0.0595	0.0017	0.0000	0.0595	-0.0131	-0.0024	0.3649	-0.0232	0.0232	0.1832
w. schema A	0.4138	0.8473	0.1968	0.3472	1.0000	0.2329	0.4432	1.0000	0.3856	0.3436	0.9360	0.4564
Δ	-0.0322	0.0745	0.1968	-0.0384	0.0000	0.2329	-0.0363	0.0000	0.3856	-0.0125	0.0648	0.4564
NSTM	0.4467	0.3120	0.6081	0.6389	0.6128	0.7905	0.6277	0.6304	0.8598	0.3569	0.2336	0.0000
w. schema K	0.4524	0.5872	0.6576	0.6596	0.7072	0.7962	0.6393	0.6824	0.8608	0.4354	0.3464	0.4881
Δ	0.0057	0.2752	0.0496	0.0206	0.0944	0.0057	0.0116	0.0520	0.0010	0.0785	0.1128	0.4881
w. schema A	0.4948	0.4978	0.6909	0.6254	0.8316	0.8286	0.6557	0.6335	0.8761	0.4683	0.3333	0.6668
Δ	0.0481	0.1858	0.0828	-0.0135	0.2188	0.0381	0.0279	0.0031	0.0163	0.1114	0.0997	0.6668
NVDM	0.3530	0.9704	0.0341	0.2885	0.9456	0.0483	0.2860	0.9552	0.0335	0.2718	0.9208	-0.0301
w. schema K	0.3265	0.9712	0.0254	0.2731	0.9528	0.0499	0.2903	0.9504	0.0327	0.2632	0.9128	-0.0282
Δ	-0.0265	0.0008	-0.0087	-0.0155	0.0072	0.0016	0.0043	-0.0048	-0.0008	-0.0086	-0.0080	0.0019
w. schema A	0.3876	0.9728	0.1001	0.3130	0.9767	0.1427	0.2910	0.9624	0.0384	0.3323	0.8933	0.1650
Δ	0.0346	0.0024	0.0660	0.0245	0.0311	0.0945	0.0050	0.0072	0.0049	0.0605	-0.0275	0.1952
PLDA	0.4155	0.8880	0.1090	0.4976	0.9056	0.1967	0.5065	0.9136	0.2984	0.3380	0.7472	0.2678
w. schema K	0.4570	0.8984	0.0908	0.3917	0.9088	0.1556	0.5147	0.9224	0.3110	0.3412	0.7680	0.3078
Δ	0.0415	0.0104	-0.0182	-0.1060	0.0032	-0.0411	0.0082	0.0088	0.0126	0.0031	0.0208	0.0400
w. schema A	0.4761	1.0000	0.2569	0.5051	0.9904	0.2758	0.4296	0.9615	0.4152	0.3401	0.9440	0.4614
Δ	0.0605	0.1120	0.1479	0.0075	0.0848	0.0791	-0.0770	0.0479	0.1168	0.0021	0.1968	0.1936

Table 1: Topic quality comparison across models and datasets using Topic Coherence(C_V), Topic Diversity (TD), and Silhouette. The Δ rows represent relative improvements over the base models, highlighting the impact of schema-guided refinement on topic coherence and structure.

Semantic quality. Despite structural improvements, C_V coherence remains competitive or improves in many cases. For example, in PLDA on 20News, C_V increases from 0.4570 to 0.4761, while maintaining high topic diversity. Overall, topic diversity is preserved or improved, suggesting that pruning removes redundancy rather than collapsing the topic space.

Consistency and robustness. Among all configurations, auto mode achieves the largest performance gains while remaining consistent across datasets and backbone models. In particular, it outperforms fixed-K settings in nearly all cases, indicating that automatic topic selection is more effective than manual specification. The improvements are especially pronounced in structural metrics, where Silhouette scores increase by up to +0.20–0.30 in absolute terms, corresponding to roughly 200%–300% relative improvement. This demonstrates that adaptively determining K not only simplifies model selection but also leads to substantially better-structured topic representations.

Dataset	Variant	CV	TD	$\Delta full(C_V)$	$\Delta full(TD)$
20NG	Vanilla	0.4800	0.7010	-0.0072	-0.0620
	Full	**0.4871**	**0.7636**	-	-
	skip-step1	0.4815	0.7458	-0.0056	-0.0178
	skip-step2	0.4822	0.7440	-0.0049	-0.0196
	skip-step3	0.4808	0.7609	-0.0063	-0.0027

Table 2: Step ablation on 20News. Each refinement step contributes to the full pipeline; removing the pruning step (skip-step2) causes the largest TD drop.

6.2 Schema Induction

To assess the effectiveness of schema induction, we analyze the refined topic organization and labeling results shown in Table 2. Using ETM as the base neural topic model, we evaluate the impact of removing each step in the LLM-based refinement pipeline on the 20NG dataset. Table 2 shows that all refinement steps contribute positively to performance. Notably, Step 3 yields the largest improve-

k	Mode	CV	TD	Silhouette
20	vanilla	0	0.850	0.213
	keep	**0.492**	0.852	0.220
	auto	0.410	**0.855**	**0.293**
30	vanilla	0.5003	0.780	0.036
	keep	**0.504**	0.791	0.113
	auto	0.400	**0.854**	**0.285**
50	vanilla	0.4602	0.6616	-0.065
	keep	0.470	0.680	-0.061
	auto	**0.496**	**0.831**	**0.198**
75	vanilla	0.4575	0.5765	-0.170
	keep	**0.462**	0.640	-0.140
	auto	0.364	**0.873**	**0.256**
100	vanilla	0.4559	0.512	-0.216
	keep	0.4566	0.5732	-0.1813
	auto	**0.496**	**0.867**	**0.222**

Table 3: Determining the optimal number of topics (k)

Method	Datasets	# of LLM Call	wall time (s)
LLM-ITL	20NG	1200	20481
	AGNews	1600	26939.7
	DBPedia	1600	28186.8
	R8	550	9724.2
TopicGPT	20NG	2178	4207
	AGNews	4988	13437.9
	DBPedia	20688	92339.8
	R8	1240	1011.9
SchemaTopic (llama-8b)	20NG	3.9	637.3
	AGNews	4.1	523.8
	DBPedia	3.3	604.7
	R8	3.5	337.5
SchemaTopic (gpt 5.2)	20NG	3.9	392.7
	AGNews	4.5	483.4
	DBPedia	4.5	445.8
	R8	4.2	429.9

Table 4: Comparison of LLM usage and wall-clock time across methods and datasets.

ment in topic coherence (CV) while providing the smallest gain in topic diversity (TD), whereas Steps 1 and 2 exhibit the opposite trend. These results suggest that the LLM-based refinement process operates in a structurally complementary manner, where different steps specialize in distinct aspects of topic quality. In particular, the pipeline progressively refines topics by first improving global organization and diversity, and then enhancing semantic coherence, leading to a well-structured and interpretable topic space.

6.3 Topic Structure

To analyze structural properties of the learned topics, we study both refinement ablations and the impact of K, as summarized in Table 3. Table 3 analyzes the impact of different topic numbers K on topic quality under vanilla, keep, and auto modes.

Auto mode achieves robust structural improvements. Across all K settings, auto mode consistently yields the highest Silhouette scores, with substantial gains over both vanilla and keep modes. For example, at $K = 50$, the Silhouette score improves from -0.065 (vanilla) and -0.061 (keep) to 0.198 (auto), and at $K = 100$, from -0.216 (vanilla) to 0.222. These correspond to large absolute gains (+0.20–0.40) and indicate that auto mode effectively restructures the topic space into well-separated clusters.

Less sensitivity to the choice of K. While vanilla and keep modes show noticeable fluctuations in performance as K varies, auto mode remains consistently strong across all settings. In particular, even when large K values degrade structural quality in baseline methods (e.g., negative Silhouette scores), auto mode maintains positive and stable scores. This suggests that automatic topic pruning mitigates over-fragmentation and reduces sensitivity to the initial choice of K.

Trade-off between structure and semantics. Although keep mode often achieves the highest C_V coherence at smaller K, auto mode provides a better overall balance by significantly improving structural quality while maintaining competitive semantic coherence. In addition, topic diversity (TD) is consistently highest under auto mode, indicating that pruning enhances diversity rather than collapsing topics.

Consistent trends across K. A clear trend is that increasing K negatively impacts structural quality in vanilla and keep modes, whereas auto mode counteracts this effect. This highlights the advantage of adaptive topic selection, showing that automatically refining the topic set leads to more stable and well-structured representations regardless of the initial K.

6.4 Cost Efficiency

Since LLM-based topic modeling incurs substantial computational cost, it is important to reduce the number of LLM calls for practical deployment. Table 4 therefore compares different methods in terms of the number of LLM calls and wall-clock time across multiple datasets. As LLM inference

dominates the overall runtime, reducing the number of LLM calls is essential for improving scalability.

LLM Calls as the Primary Bottleneck. Across all baselines, we observe a strong correlation between the number of LLM calls and the wall-clock time. In particular, methods that rely on frequent LLM invocation incur substantial computational overhead. For example, TopicGPT requires up to 20,688 LLM calls on DBPedia, resulting in 92,339.8 seconds of runtime. Similarly, LLM-ITL performs thousands of calls (e.g., 1,600 on DBPedia), leading to high latency (28,186.8 seconds). These results clearly indicate that repeated LLM invocation constitutes the dominant computational bottleneck.

7 Conclusion

We presented **SchemaTopic**, a framework that combines neural topic models (NTMs) with LLM-based schema induction to produce more coherent, diverse, and structurally organized topics. Our approach operates in three phases: (1) **warm-up training**, where an initial topic space is learned; (2) **LLM-based topic refinement**, where an LLM induces a semantic schema by identifying a partitioning criterion, pruning low-quality topics, and assigning schema labels; and (3) **anchor-guided retraining**, where the refined schema is used to retrain the NTM with an anchor-guided loss.

This design addresses key limitations of prior work. SchemaTopic introduces schema-level organization beyond flat topic lists, enables adaptive topic selection via pruning, and reduces LLM cost by operating only on topic-level summaries.

Experiments on multiple datasets and NTM backbones show consistent improvements in topic coherence (Röder et al., 2015) and diversity (Dieng et al., 2020b). We believe this schema-guided approach offers an effective and scalable way to integrate NTMs with LLMs.

8 Limitations

Our work has several limitations that suggest directions for future research.

First, the quality of the induced schema depends on the LLM's ability to reason over topic-word lists, which may be sensitive to the choice of LLM and prompt design. Although we demonstrate consistent improvements with both open-source and proprietary LLMs, the schema induction quality could degrade for highly specialized or low-resource domains where the LLM lacks sufficient background knowledge.

Second, while SchemaTopic automatically adjusts the effective number of topics through pruning, the initial topic count K still needs to be specified for the warm-up NTM. The final topic quality may vary depending on this initial setting, and exploring adaptive initialization strategies remains an open problem.

Third, the current framework induces a single flat schema with one partitioning criterion. Extending this to multi-level or multi-faceted schemas could better capture complex thematic structures in large-scale corpora.

Finally, our evaluation relies on standard automatic metrics such as C_V coherence (Röder et al., 2015) and topic diversity (Dieng et al., 2020b). These metrics, while widely used, do not fully capture the interpretability and practical utility of schema-organized topics from a human perspective. Incorporating human evaluation would provide a more comprehensive assessment of the framework's effectiveness.

References

Anthropic. 2024. Claude 3 technical report. *arXiv preprint*.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Victor Bulatov, Vasiliy Alekseev, and Konstantin Vorontsov. 2024. Determination of the number of topics intrinsically: Is it possible? *arXiv preprint arXiv:2406.10402*.

Chia-Hsuan Chang, Yi-Lun Hsieh, and 1 others. 2025. Lita: Llm-assisted iterative topic augmentation. *arXiv preprint arXiv:2412.12459*.

Shuyu Chang, Rui Wang, Peng Ren, and Haiping Huang. 2024. Enhanced short text modeling: Leveraging large language models for topic refinement. *arXiv preprint arXiv:2403.17706*.

Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. 2020a. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. 2020b. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*.

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

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *International Conference on Learning Representations*.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. *Proceedings of ICML*.

Jens Lehmann and 1 others. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*.

Yishu Miao, Edward Grefenstette, and Phil Blunsom. 2017. Discovering discrete latent topics with neural variational inference. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2410–2419.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *ICML*.

Yuji Miyamoto and Ryohei Sasano. 2023. Dynamic neural topic model with topic hierarchy. In *Findings of ACL 2023*, pages 12062–12074.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Chau Minh Pham, Alexander Hoyle, Simeng Sun, Philip Resnik, and Mohit Iyyer. 2024. Topicgpt: A prompt-based topic modeling framework. In *Proceedings of NAACL 2024*, pages 2956–2984.

Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*.

Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *WSDM*.

Kyle Schroeder and 1 others. 2025. Reliability of topic modeling. In *Proceedings of NAACL 2025*.

Akash Srivastava and Charles Sutton. 2017a. Autoencoding variational inference for topic models. In *International Conference on Learning Representations*.

Akash Srivastava and Charles Sutton. 2017b. Autoencoding variational inference for topic models. In *ICLR*.

Hugo Touvron and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Xiaobao Wu, Xinshuai Dong, Thong Nguyen, and Anh Tuan Luu. 2023a. Effective neural topic modeling with embedding clustering regularization. In *International Conference on Machine Learning*.

Xiaobao Wu, Xinshuai Dong, Thong Nguyen, and Anh Tuan Luu. 2023b. Effective neural topic modeling with embedding clustering regularization. In *Proceedings of the 40th International Conference on Machine Learning*, pages 37335–37357. PMLR.

Xiaobao Wu, Thong Nguyen, and Anh Tuan Luu. 2024a. A survey on neural topic models: Methods, applications, and challenges. *Artificial Intelligence Review*.

Xiaobao Wu and 1 others. 2024b. On the affinity, rationality, and diversity of hierarchical topic modeling. In *Proceedings of AAAI*.

Xiaohao Yang, He Zhao, Weijie Xu, Yuanyuan Qi, Jueqing Lu, Dinh Phung, and Lan Du. 2025. Neural topic modeling with large language models in the loop. In *Proceedings of ACL 2025*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*.

He Zhao and 1 others. 2021. Neural sparse topic modeling. In *ACL*.

Appendix

A Additional Experimental Results

	20NG			AGNews			DBPedia			R8		
	CV	**TD**	**PN**	**CV**	**TD**	**PN**	**CV**	**TD**	**PN**	**CV**	**TD**	**PN**
NVDM	0.352	0.970	0.162	0.288	0.945	0.254	0.285	0.955	0.263	0.271	0.920	0.407
w. keep	0.345	0.976	0.157	0.288	0.965	0.254	0.295	0.965	0.270	0.275	0.928	0.402
	-1.989%	0.619%	-3.086%	0.000%	2.116%	0.000%	3.509%	1.047%	2.662%	1.476%	0.870%	-1.229%
w. auto	0.337	0.974	0.157	0.288	0.973	0.291	0.291	0.964	0.265	0.284	0.912	0.383
	-4.261%	0.412%	-3.086%	0.000%	2.963%	14.567%	2.105%	0.942%	0.760%	4.797%	-0.870%	-5.897%
PLDA	0.415	0.888	0.449	0.497	0.905	0.511	0.506	0.913	0.700	0.338	0.747	0.661
w. keep	0.429	0.908	0.458	0.553	0.936	0.512	0.520	0.924	0.711	0.337	0.818	0.668
	3.373%	2.252%	2.004%	11.268%	3.425%	0.196%	2.767%	1.205%	1.571%	-0.296%	9.505%	1.059%
w. auto	0.391	0.855	0.443	0.498	0.932	0.494	0.443	0.889	0.716	0.318	0.775	0.644
	-5.783%	-3.716%	-1.336%	0.201%	2.983%	-3.327%	-12.451%	-2.629%	2.286%	-5.917%	3.748%	-2.572%
ETM	0.479	0.701	0.405	0.462	0.783	0.557	0.486	0.796	0.753	0.393	0.609	0.679
w. keep	0.483	0.749	0.407	0.466	0.864	0.547	0.506	0.861	0.758	0.404	0.692	0.670
	0.835%	6.847%	0.494%	0.866%	10.345%	-1.795%	4.115%	8.166%	0.664%	2.799%	13.629%	-1.325%
w. auto	0.513	0.822	0.371	0.483	0.852	0.552	0.517	0.854	0.761	0.413	0.754	0.686
	7.098%	17.261%	-8.395%	4.545%	8.812%	-0.898%	6.379%	7.286%	1.062%	5.089%	23.810%	1.031%
NSTM	0.437	0.286	0.373	0.638	0.612	0.576	0.627	0.630	0.775	0.466	0.233	0.682
w. keep	0.446	0.312	0.361	0.645	0.726	0.575	0.640	0.703	0.776	0.469	0.285	0.680
	2.059%	9.091%	-3.217%	1.097%	18.627%	-0.174%	2.073%	11.587%	0.129%	0.644%	22.318%	-0.293%
w. auto	0.468	0.474	0.354	0.649	0.665	0.577	0.644	0.638	0.778	0.467	0.312	0.678
	7.094%	65.734%	-5.094%	1.724%	8.660%	0.174%	2.711%	1.270%	0.387%	0.215%	33.906%	-0.587%
ECRTM	0.446	0.772	0.490	0.385	1.000	0.517	0.479	1.000	0.782	0.356	0.871	0.662
w. keep	0.441	0.736	0.481	0.396	1.000	0.505	0.480	0.998	0.755	0.357	0.872	0.659
	-1.121%	-4.663%	-1.837%	2.857%	0.000%	-2.321%	0.209%	-0.200%	-3.453%	0.281%	0.115%	-0.453%
w. auto	0.437	0.842	0.477	0.375	1.000	0.525	0.441	1.000	0.733	0.362	0.820	0.659
	-2.018%	9.067%	-2.653%	-2.597%	0.000%	1.547%	-7.933%	0.000%	-6.266%	1.685%	-5.855%	-0.453%

Table 6: Topic quality comparison across model(gpt 5.2) and datasets using Topic Coherence(CV), Topic Diversity (TD), and Silhouette. The rows represent relative improvements over the base models, highlighting the impact of schema-guided refinement on topic coherence and structure.