Platform-Invariant Topic Modeling via Contrastive Learning to Mitigate Platform-Induced Bias

Minseo Koo[1], **Doeun Kim**[1], **Sungwon Han**[2]*, **Sungkyu Park**[3]*
[1]Department of AI Convergence, Kangwon National University, South Korea
[2]School of Computing, KAIST, South Korea
[3]KDI School of Public Policy and Management, South Korea
{ms4563321, kde9867}@kangwon.ac.kr, lion4151@kaist.ac.kr, shaun@kdischool.ac.kr

Abstract

Cross-platform topic dissemination is one of the research subjects that delved into media analysis; sometimes it fails to grasp the authentic topics due to platform-induced biases, which may be caused by aggregating documents from multiple platforms and running them on an existing topic model. This work deals with the impact of unique platform characteristics on the performance of topic models and proposes a new approach to enhance the effectiveness of topic modeling. The data utilized in this study consisted of a total of 1.5 million posts collected using the keyword "ChatGPT" on the three social media platforms. The devised model reduces platform influence in topic models by developing a platform-invariant contrastive learning algorithm and removing platform-specific jargon word sets. The proposed approach was thoroughly validated through quantitative and qualitative experiments alongside standard and state-of-the-art topic models and showed its supremacy. This method can mitigate biases arising from platform influences when modeling topics from texts collected across various platforms.

1 Introduction

Topic modeling is a well-known method in the realm of Natural Language Processing (NLP) to identify coherent topics within a text corpus. Topic modeling is used in a variety of areas, including trends and major events on social media (Park et al., 2021b; Curiskis et al., 2020). In the realm of media analysis, researchers often analyze cross-platform topic (or information) dissemination to understand general trends on social media. Their general approach is to aggregate posts, i.e., documents, from multiple platforms rather than analyzing each platform individually (Danescu-Niculescu-Mizil et al., 2013; Park et al., 2021a), thereby being exposed to platform-induced bias. In that sense, it is beneficial to devise an integrated topic modeling when investigating the influence of media platforms.

There are two ways to extract topics from documents: counting word frequency to determine topics or clustering words with similar meanings. These methods correspond to the Bag of Words (hereafter, BoW) approach and the contextual approach, respectively.

Algorithms based on BoW assume that the words within a document follow a Dirichlet distribution and extract the topic distribution within the document (Miao et al., 2016; Yan et al., 2013; Fisher et al., 2020). Latent Dirichlet Allocation (LDA) is one of the representative models (Blei et al., 2003). There is also a large body of research on topic modeling methodologies based on LDA (Zhao et al., 2017; Nan et al., 2019; Blei et al., 2010). Conversely, context-based topic modeling algorithms cluster elements with similar contexts before producing topics (Thompson and Mimno, 2020; Xun et al., 2017). BERTopic, which uses the pre-trained model BERT, is an example of context-based topic modeling (Grootendorst, 2022).

With posts collected from multiple social media platforms, topic modeling algorithms can be an efficient tool to analyze and understand the predominant topics of discussion (Godin et al., 2013; Cha and Cho, 2012; Taecharungroj, 2023). However, when conducting topic modeling on various internet social media platforms, the unique characteristics of each platform can influence the topic modeling algorithms. These platform-specific characteristics could be differences in writing styles, i.e., semantic traits, within posts on the platform, or they could be inherent features, i.e., syntactic traits, of the platform's functionality. Regarding semantic traits, for example, X (i.e., Twitter) focuses on quick communication through short messages, Reddit is more suited for writing explanatory posts, while YouTube is characterized by comments on

*Corresponding Authors

Findings of the Association for Computational Linguistics: EACL 2024, pages 11123–11139

videos (Choi et al., 2016). Due to these differences, topics extracted in topic modeling can vary inconsistently across platforms. Regarding syntactic traits, on the other hand, platform-specific jargon (hereafter referred to as jargon), such as 'RT' and 'retweet' predominantly used on X, can influence the results of topic modeling algorithms. If such jargon is included in the process, algorithms might be biased toward extracting topics specific to a particular platform. This study first aims to answer the following questions through preliminary studies:

- RQ1: Do different platforms have unique platform characteristics? (*Yes*)
- RQ2: Are topic modeling algorithms influenced differently by platforms? (*Yes*)

Based on these insights, we introduce Platform-Invariant Topic modeling (PITopic), designed to nullify the influence of platform-specific characteristics. By removing jargon and applying a new platform-invariant contrastive learning algorithm, PITopic ensures that the extracted topics are universally coherent, irrespective of the platform. Experiments with both real-world and synthetic datasets with multiple platforms (or sources) confirm our model's superiority over contemporary baselines in both topic diversity and coherence.

The code and implementation details for the model can be found in a GitHub repository. [1]

2 Related Works

There are largely three streams of topic modeling: BoW-based methods, clustering methods, and methods that combine these two methods.

LDA (Blei et al., 2003) is a typical BoW-based algorithm that assumes a Dirichlet before the topic distribution. It follows a probabilistic Latent Semantic Analysis (pLSA) assumption that all word tokens in a document are sampled from a mixture of latent topics (Hoffman, 1999).

BoW-based algorithms do not consider the order of words in sentences when extracting topics from documents. This leads to a limitation in topic modeling with platform data: the frequency of simultaneous word occurrences can be low and influenced by words predominantly used on a specific platform. To address these limitations of LDA, autoencoding variational Bayes (AEVB) can be applied to topic modeling (Srivastava and Sutton, 2017). The AVITM method trains inference networks that map approximate posterior distributions directly from documents. However, this method relies on the quality of the training data, making it difficult to learn the exact posterior distribution in the presence of noise, such as individual characteristics of platform data. Moreover, both methods cannot capture the contextual information of sentences in long text documents.

BERTopic effectively embeds sentences using the pre-trained model BERT (Devlin et al., 2019) and extracts important topic words from documents through the clustering and c-TF-IDF algorithm (Grootendorst, 2022). Yet, this clustering-based method has limitations in representing situations where multiple topics can coexist simultaneously. Recently, the Contextualized Topic Model (CTM) was introduced to apply BERT (Devlin et al., 2019) for extracting the embedding vector and reconstructing BoW for each sentence (Bianchi et al., 2021) with Variational Autoencoder (VAE) (Kingma and Welling, 2013). Since the model still uses BoW, it can be influenced by words predominantly used on a single platform, and unique platform characteristics in BERT embeddings can yield biased topic modeling results.Recent research combines the advantages of both BoW-based and clustering methods for topic modeling. This approach uses contrastive learning and term weighting with topic word extraction to effectively integrate these benefits (Han et al., 2023). When multiple platforms exist, however, the extraction of topic words can become ambiguous due to platform characteristics.

In this research, we propose a method to overcome the weakness that existing topic modeling algorithms rely on the influence of platforms and effectively extract topics by utilizing various platform data. This research seeks to reduce the influence of platforms on topic modeling and ensure the performance and consistency of topic modeling algorithms.

3 Preliminary Studies

We here provide preliminary studies to answer our research questions: (RQ1) whether different platforms have their unique characteristics, and (RQ2) these traits differently influence the performance of topic modeling algorithms.

Dataset. For this study, data was collected directly from three platforms: X, Reddit, and

[1] https://github.com/kde9867/Platform-Invariant-Topic-Modeling

YouTube, respectively (Figure 1). We used a keyword to collect data so that we could see how each platform reacted to one topic, and for the current work, we set the keyword to "ChatGPT". Note that the keyword was arbitrarily chosen to allow the platforms to bring up topics within a specific domain. We collected only English-language data with the keyword from December 1, 2022, to March 2, 2023. 250,000 X tweets, Reddit posts, and YouTube comments were collected. The aggregated real-world data was used in the preliminary studies and the main experiments.

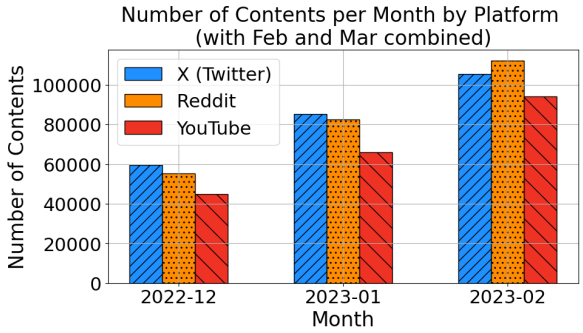

Figure 1: Distribution of data counts by month: sampled at the same size (250,000) per platform, with March 1-2 data combined with February.

RQ1. Do different platforms have unique platform characteristics? (*Yes*)

H1-1: Topics were distributed differently between platforms – *Rejected.*

For each platform, we extracted 20 topics using LDA, and two human annotators decided on the topic names based on the top topic words corresponding to each topic. We then randomly sampled 2,000 data for each platform and called the ChatGPT API to label the sampled data from the decided topic names, i.e., we conjectured ChatGPT as a human annotator and made it distribute the topics. We used JS Divergence to compare the topic distribution between and within platforms, as presented in Table 1: note that we randomly sampled 100 labeled results from each platform and calculated JS Divergence 100 times, then computed an average. We confirmed that topic distributions between platforms were not so different than those within platforms, so we rejected the null hypothesis.

H1-2: There are no unique characteristics in text across platforms – *Rejected.*

To determine if there are unique characteristics

Platform	JS Divergence
X-Reddit (Between)	0.227
X-YouTube	0.244
Reddit-YouTube	0.215
X (Within)	0.209
Reddit	0.209
YouTube	0.213

Table 1: Comparison of JS Divergence of topic distributions between and within platforms.

across platforms, we experimented with predicting platform origin using the collected data. Suppose we can accurately predict the origin of a text by learning the differences in unique characteristics across platforms. In that case, it means that each platform's characteristics are distinct. We labeled X data as 0, Reddit as 1, and YouTube as 2. We then divided the text into specific units (X: tweets, Reddit: posts, YouTube: comments) and fed them into the RoBERTa model to predict platform origin (Liu et al., 2019). The results showed an accuracy rate of 96% on the unseen test set, suggesting that characteristics are distinct across platforms, so we rejected the null hypothesis.

To sum up, two hypothesis testing results show that while the actual topic distributions are not so different across platforms, there are latent unique characteristics for each platform.

RQ2. Are topic modeling algorithms influenced differently by platforms? (*Yes*)

H2: Mutual Information between topics and platforms are similar across topic models – *Rejected.*

Mutual Information (hereafter MI) between topics and platforms was compared to see if topics are determined by platforms during the topic modeling process (Eq. 1, see in Appendix A.2 for the full equation). The models LDA and BERTopic were adopted as representative models of the BoW-based and the clustering method, respectively.

MI is a measure of how similar the joint distribution $p(X, Y)$ is to $p(X)p(Y)$. In this study, $p(X)$ was set to the overall probability of the platform in question, $p(Y)$ to the probability of the overall topic, and $p(X, Y)$ to the probability of having a specific topic label number on the platform in question, and MI between platforms and topics was calculated. Furthermore, we looked at conditional entropy $H(T|P)$, which allows us to see how uncertainly the topic modeling

algorithm determines topic (T) under the influence of platform (P). If the topic modeling algorithm assigns all documents to a single topic under the strong influence of the platform, then the conditional entropy value will be zero. This study also compared the entropy of topics to see how uniformly the topic algorithm assigns topics.

$$I(P;T) = H(T) - H(T|P) \quad (1)$$

	MI	$H(T)$	$H(T\|P)$
BERTopic	0.085	1.281	1.197
LDA	0.100	2.840	2.739
Annotations	0.105	3.317	3.212

Table 2: Comparison of statistics (MI, entropy, and conditional entropy) between topics and platforms.

As shown in Table 2, the results for MI, $H(T)$, and $H(T|P)$ measure the extent to which the BoW (i.e., LDA) and context-based (i.e., BERTopic) topic modeling algorithms are affected by platforms. Comparing the MI of all models, we find that BERTopic has the smallest MI value compared to other models, while LDA and Annotations are similar. On the other hand, when we compared conditional entropy $H(T|P)$ between LDA and BERTopic, BERTopic has a lower value than LDA, which means that in the case of BERTopic, the topics are not extracted uniformly due to the platform and are relatively more clustered on certain topics. We confirmed that each topic modeling algorithm is affected by the platform in its execution results, so we rejected the null hypothesis. Interestingly, when compared with topic annotations made via ChatGPT (i.e., Annotations in Table 2), all topic models have a comparatively lower $H(T|P)$, indicating that the platform bias significantly affects the topic solutions from models.

Given the answers to the two RQs, we acknowledge the necessity of developing a topic model to discard platform characteristics, especially when data from multiple sources are combined.

4 Platform-Invariant Topic Modeling

Problem formulation: Let $\mathcal{P}_{\text{all}} = \{\mathcal{P}_j\}_{j=1}^{m}$ be a set of platforms, each containing a collection of documents $\mathcal{X}_j$. Given an input document $\mathbf{x} \in \mathcal{X}_j$, the primary goal of our topic model is to estimate and assign a topic from the predefined set of K topics. From the model, we extract the top-M words representing each topic to identify the theme of the topic and measure its classification quality.

We introduce PITopic, a neural topic model that extracts coherent topics from documents collected across various platforms while avoiding bias towards any specific platform. Figure 2 illustrates our concept. Our model employs a traditional pLSA-based approach (Hoffman, 1999), estimating the topic distribution from the text and then reconstructing the BoW distribution of the original input from the estimated topic distribution. To eliminate bias introduced by platform grouping, we propose two components. First, during the process of estimating the topic distribution, we use contrastive learning to optimize the conditional mutual information of two similar text pairs concerning the platform, preventing the topic from being predominantly drawn from a single platform (Section 4.1). Second, in reconstructing the BoW distribution from the topic distribution, we minimize platform-dependent jargon in composing the BoW, thereby removing bias (Section 4.2). These two methods enable our model to handle biases from both the unique textual styles and different word usage patterns inherent in each platform. Each component is outlined in the following section.

4.1 Platform-Invariant Contrastive Learning

In this component, we utilize contrastive learning to remove the bias imparted by platforms in topic embedding. Given a text sample $\mathbf{x}$ from batch $\mathcal{B}$, we first encode the data using a pre-trained and fixed language model backbone (e.g., BERT (Devlin et al., 2019)) f. Then, the encoded representation is mapped to a topic embedding through a shallow neural network model g (Eq. 2). The topic embedding $\mathbf{z}$, having the same dimension as the number of topics K in the pre-defined topic set, is transformed into a topic probability vector $\mathbf{t}$ through the softmax function, indicating the probability of the sample being assigned to each topic.

$$\mathbf{z} = g \odot f(\mathbf{x}), \quad \mathbf{t} = \text{SoftMax}(\mathbf{z}) \quad (2)$$

PITopic incorporates a generalized contrastive learning objective to learn the topic distribution (Chen and Li, 2020; Wang and Isola, 2020). This objective consists of an alignment loss and a distribution loss. The alignment loss enhances the cosine similarity of the topic probability vectors of the nearest neighbors based on the pre-trained language model embeddings (L_{align}), assuming

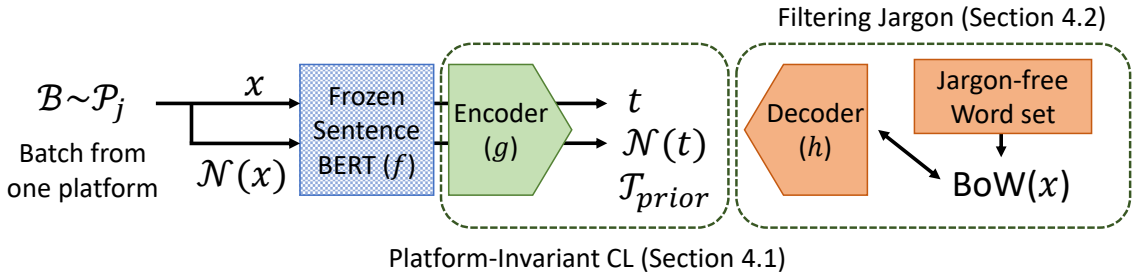

Figure 2: The illustration of PITopic. PITopic consists of two main components to reduce platform bias in topic modeling: platform-invariant contrastive learning and filtering jargon for BoW reconstruction.

that the semantically similar sample pairs should have the same topic. On the other hand, the distribution loss regularizes the topic probability distribution $\mathcal{T}$ within the batch $\mathcal{B}$ to follow a prior Dirichlet distribution $\mathcal{T}_{\text{prior}}$ (L_{dist}), which prevents the model from finding trivial solutions: assigning the same topic for all samples. Sliced Wasserstein Distance (SWD) measures the distance between two distributions (Kolouri et al., 2019). The formulation of each loss is defined as follows.

$$L_{\text{align}} = -\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \text{sim}(\mathbf{t}^i, \mathcal{N}(\mathbf{t}^i)) \quad (3)$$

$$L_{\text{dist}} = \text{SWD}(\mathcal{T}, \mathcal{T}_{\text{prior}}), \quad (4)$$

where the superscript i represents the index of the sample in the batch, sim$(\cdot)$ measures the cosine similarity between two input vectors and, $\mathcal{N}(\cdot)$ is the function that produces the top-1 nearest neighbor of i over the language model's embedding space.

Here, we design a strategy for selecting batches from each platform dataset to minimize platform bias in the contrastive objective. Consider a scenario where the samples in a batch are assumed to come solely from one platform (i.e., $\mathcal{B} \subset \mathcal{X}_j$). In this case, minimizing the distribution loss would match the topic distribution of that specific platform to the prior Dirichlet distribution $\mathcal{T}_{\text{prior}}$. If this process is repeated for each platform separately, meaning that the samples in each batch always come from the same platform, then the topic probability distribution created by all platforms would conform to the same prior Dirichlet distribution $\mathcal{T}_{\text{prior}}$. As a result, it becomes difficult to infer the platform from the sample's topic embedding. This is akin to maximizing the mutual information between two positive samples conditioned on the platform, thereby extracting features of topics that are independent of the platform (Ma et al., 2021). The total loss for contrastive learning is the following:

$$L_{\text{contrastive}} = L_{\text{align}} + L_{\text{dist}}. \quad (5)$$

4.2 Filtering Jargon for BoW Reconstruction

In the next step, our proposed topic model utilizes topic probabilities to reconstruct the BoW representation of a given text sample. Typically, when constructing the BoW representation from a text sample, instead of using the entire set of words, we select those most likely to contribute to the topic. For instance, measures like TF-IDF are often employed to preferentially use words that appear frequently in a specific document set (Ramos et al., 2003). However, a challenge arises due to the collection of texts from various platforms forming the corpus; such measures can be significantly influenced by the jargon used within individual platforms. To mitigate this risk, this component introduces a process of preemptively removing jargon when constructing the BoW.

Given the set of all platforms $\mathcal{P}_{all}$, a word that appears frequently in one platform but rarely in others can be considered platform-dependent jargon. To identify such words, we modify the traditional definition of TF-IDF, similar to (Grootendorst, 2022). Rather than measuring term frequency (TF) and inverse document frequency (IDF) at the document level, we calculate these for all documents from a single platform combined as one unified document by following the c-TF-IDF concept. This approach allows us to assess how frequently a word appears (using TF) and how concentrated it is in a single platform (using IDF). The product of these two measures (jargon score) is used to list words in descending order. Subsequently, we remove the top-N likely jargon words from the word set of

each platform before constructing the BoW.

$$\text{tf}(w, \mathcal{P}_j) = \frac{\text{count}(w, \mathcal{P}_j)}{\sum_{w' \in \mathcal{P}_j} \text{count}(w', \mathcal{P}_j)} \quad (6)$$

$$\text{idf}(w, \mathcal{P}_{all}) = \log \frac{|\mathcal{P}_{all}|}{|\mathcal{P}_j \in \mathcal{P}_{all} : w \in \mathcal{P}_j|} \quad (7)$$

$$\text{jargon-score}(w) = \text{tf}(w, \mathcal{P}_j) \cdot \text{idf}(w, \mathcal{P}_{all}), \quad (8)$$

where count$(\cdot)$ is a function that counts the occurrences of a word within a platform.

When the estimated topic distribution $\mathbf{t}$ of an input sample $\mathbf{x}$ is given, the model is trained through the following loss to reconstruct the jargon-filtered BoW representation using a decoder h.

$$L_{\text{recon}} = -\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \text{BoW}(\mathbf{x}^i) \cdot \log h(\mathbf{t}^i), \quad (9)$$

where the BoW$(\cdot)$ function transforms the input text into a BoW representation expressed as a probability vector. The total loss for training is set as the sum of the contrastive objective and the reconstruction objective: $L_{\text{total}} = L_{\text{contrastive}} + L_{\text{recon}}$. For inference, we input a sentence into the model and assign the topic with the highest probability in $\mathbf{t}$ to the sentence.

5 Experiments

5.1 Evaluation on Real-World Data

To evaluate our model's effectiveness in discovering coherent topics from multiple platforms, we first use the real-world dataset mentioned in our preliminary studies (Section 3).

Evaluation metrics. Four evaluation metrics are employed: MI, Topic diversity, NPMI, and UCI. MI measures the mutual dependence between the topic and platform labels (see Eq. 1). The specifics of the remaining metrics are detailed below.

- Topic diversity measures the diversity of top-M words across topics (Dieng et al., 2019). Diversity is measured by the ratio of unique words in the aggregated top-M word sets from all topics (i.e., the union of the top-M word sets without duplicates / (2 * M)).
- NPMI measures the co-occurrence probability of word-pairs among the top-M terms of a topic (Aletras and Stevenson, 2013). Higher values imply a strong semantic relationship among top-M terms.
- UCI measures the mutual information of all word pairs within the top-M words, where the word probabilities and co-occurrences are computed from text within a sliding window, iterating over the reference corpus (Newman et al., 2010).

Unlike other topic modeling works, we measured topic coherence for *each platform separately*, rather than across the entire dataset, to verify if our topic model can extract coherent topics independently of the platform. If the model generates topics related only to a specific platform, then the coherence of such topics would significantly decrease when applied to other platforms. Consequently, as the number of these platform-specific topics increases, the overall topic coherence measure for each platform also decreases. On the other hand, if the model extracts platform-invariant topics, then high topic coherence will be observed across all platforms.

Implementation details. We employed the pre-trained language model Sentence-BERT, specifically the allMiniLM-L6-v2 variant (Reimers and Gurevych, 2019), for our analysis. The number of topics K was set to 20. The Dirichlet prior for $\mathcal{T}_{\text{prior}}$ was set to 0.05 (i.e., 1/# of topics). The size of the BoW vocabulary was set to 30,000. The number of jargon filtered N was set to 100 per platform (i.e., a total of 300). Due to space limitations, full details are described in Appendix A.3.

Baselines. We use seven baselines for comparison. The baselines used were: (1) LDA (Blei et al., 2003), which uses BoW to extract topics over the pLSA assumption; (2) AVITM (Srivastava and Sutton, 2017), which enhances LDA by incorporating Autoencoded Variational Inference; (3) Top2vec (Angelov, 2020), which utilizes Doc2Vec (Le and Mikolov, 2014) representations to embed words and documents to identify topic clusters; (4) ClusterTM (Sia et al., 2020), which clusters pre-trained word embeddings based on document information, facilitating the reorganization of keywords for topic identification; (5) Contextualized Topic Model (CTM) (Bianchi et al., 2020), where document contexts are captured to enrich the topic modeling process; (6) BERTopic (Grootendorst, 2022), where BERT-based sentence embeddings are used in conjunction with clustering algorithms to discover topics; (7) UTopic (Han et al., 2023), which integrates BoW with clustering techniques, optimizing document embeddings via con-

Real-World data

Model	Mutual Information	Topic Diversity	NPMI				UCI			
			X	Reddit	YouTube	Average	X	Reddit	YouTube	Average
LDA	0.097	0.695	0.325	0.249	0.188	0.254	0.985	1.409	1.315	1.236
AVITM	**0.006**	0.452	0.080	0.247	0.120	0.149	0.491	0.981	0.676	0.716
Top2Vec	0.490	0.567	0.150	0.264	0.239	0.218	1.200	1.958	1.854	1.671
ClusterTM	0.086	0.524	0.100	0.226	0.156	0.161	0.694	0.981	0.932	0.869
CTM	0.187	0.839	0.200	0.354	0.203	0.252	1.473	1.984	1.480	1.746
BERTopic	0.073	0.957	0.322	0.363	0.330	0.339	2.612	2.610	2.595	2.606
UTopic	0.299	0.637	0.097	0.208	0.129	0.198	0.693	1.106	0.855	0.885
PITopic (Ours)	0.032	**0.973**	**0.450**	**0.684**	**0.453**	**0.529**	**3.571**	**3.479**	**3.626**	**3.559**

Synthetic data

Model	Mutual Information	Topic Diversity	NPMI				UCI			
			20NewsGroup	BBC	New York Times	Average	20NewsGroup	BBC	New York Times	Average
LDA	0.404	0.593	0.232	0.247	0.380	0.286	0.900	1.064	1.876	1.280
AVITM	**0.018**	0.702	0.234	0.252	0.442	0.309	1.073	1.244	2.304	1.540
Top2Vec	0.881	0.637	0.443	0.384	0.625	0.484	2.180	1.814	3.363	2.452
ClusterTM	0.462	0.702	0.290	0.309	0.432	0.344	1.325	1.380	2.213	1.639
CTM	0.372	0.814	0.422	0.298	0.597	0.439	2.179	1.110	3.207	2.166
BERTopic	0.551	0.903	0.346	0.353	0.482	0.393	1.674	1.698	2.512	1.961
UTopic	0.639	0.804	0.349	0.332	0.456	0.379	1.655	1.513	2.327	1.832
PITopic (Ours)	0.052	**0.982**	**0.500**	**0.462**	**0.656**	**0.539**	**2.552**	**2.318**	**3.415**	**2.762**

Table 3: Performance evaluation summaries over real-world data (Above) and synthetic data (Below). "Average" in the table indicates the averaged topic coherence results across all platforms. Results are averaged across different number of topics - 10, 20, and 30. Full results can see in Appendix A.4. Best performances are marked in bold.

trastive learning for improved topic representation.

Result. Table 3 (above) summarizes the performance comparison results using real-world data. We averaged the results from different numbers of topics - 10, 20, and 30; full results are presented in Table 6, 7 in Appendix A.4. PITopic showed superior performance in both topic diversity and topic coherence when compared to other baselines, while showing a lower dependency on platforms as indicated by MI. Our proposed model consistently extracted topic words with high topic coherence across all platforms, in contrast to cases like AVITM and CTM, which produced topics of relatively lower quality on specific platforms (e.g., AVITM & CTM results over X data). Note that, although AVITM shows lower MI than ours, it significantly falls short in terms of topic quality, such as coherence or diversity, compared to our model.

5.2 Evaluation on Synthetic Data

In addition to the real data we collected, we created a synthetic dataset by combining multiple benchmark datasets from different news platform sources that share the same topics. Subsequently, we evaluated whether our model could still extract coherent topics that are independent of the unique feature biases associated with each platform.

Synthetic data details. We utilized three news datasets for our analysis: 20NewsGroup (Mitchell, 1999), BBC (Greene and Cunningham, 2006), and The New York Times (Alexander, 2023), each comprising 11,314, 1,329, and 16,787 posts, respectively. All datasets contain unique topic labels, from which we extracted three topics shared across all platforms (i.e., Tech, Sports, and Politics), and compiled the corresponding documents into a single dataset. The resulting dataset contains a total of 3,600 samples.

Result. Table 3 (below) presents the performance comparison results over synthetic data. Compared to baselines, we again observed that our model is capable of discovering more universal irrespective of platforms, diverse, and coherent topics over the synthetic benchmark dataset which is composed of mixed content from various platforms.

5.3 Ablation Studies

PITopic contains two key components: platform-invariant contrastive learning and jargon filtering. To evaluate the impact of each component on performance, we conducted a study comparing four different ablations. The first three ablations involved variations of contrastive loss, removing alignment loss, distribution loss, or both. The final ablation retained jargon within the BoW. Note that

	Mutual Information	Topic diversity	NPMI X	NPMI Reddit	NPMI YouTube	NPMI Average	UCI X	UCI Reddit	UCI YouTube	UCI Average
Full model	0.028	0.955	0.526	0.767	0.644	0.646	4.529	3.551	5.155	4.412
Without L_{align} in Eq. 3	0.042	0.990	0.660	0.613	0.485	0.586	5.459	5.229	5.016	4.901
Without L_{dist} in Eq. 4	0.033	0.990	0.354	0.415	0.299	0.356	2.848	2.443	2.531	2.607
Without $L_{\text{contrastive}}$ in Eq. 5	0.091	0.970	0.501	0.368	0.443	0.437	4.198	3.003	3.626	3.609
BoW including jargon	0.040	0.975	0.450	0.541	0.232	0.408	3.787	3.068	1.647	2.834

Table 4: Ablation study results over the real-world dataset. The number of topics was set to 20. Results indicate that all components contribute to improving the topic quality.

we mainly set 20 numbers of topics as a default hyper-parameter for the remaining experiments as it reported the best topic coherence.

Table 4 summarizes the results, comparing the performance across the ablations. As expected, the full model with all components recorded higher average topic coherence than the others. In particular, removing the contrastive loss resulted in a more pronounced imbalance in topic coherence across platforms, suggesting that biases inherent to each platform directly influence coherence. Additionally, removing jargon enhanced overall performance, underscoring the effectiveness of jargon filtering in improving topic model quality. In Appendix A.4, we further detail hyper-parameter analysis on the number of jargon filtered.

Also, to confirm if, besides the jargon filtering process, our model structure depicted in Figure 2 is effective, we compared ours with other topic models injecting jargon-filtered data. PITopic yields greater performance than others as presented in Table 8 in Appendix. Together with Table 4 and 8 suggests that jargon filtering and model structure are both crucial to the integrated topic modeling.

5.4 Qualitative Analysis

Jargon analysis. Here, we confirm that PITopic effectively removes platform-specific bias through qualitative analysis. PITopic employs c-TF-IDF to filter out jargon within each platform. The identified jargon is words predominantly circulated within a single platform, such as 'remindme' and 'giphy' in Reddit, which are related to platform-specific functionalities, or 'zronx' and 'jontron' in YouTube, referring to specific broadcasters (see Table 9 in Appendix displaying the words with the top-100 jargon scores (Eq. 8) for each platform). These jargon words do not contribute to the discovery of coherent topics, thus underscoring the utility of our methodology.

Topic Conformance. Each topic's top M words derived from PITopic are listed in Table 11 (top) in Appendix, which shows the semantic consistency among the retrieved words as explained in Appendix A.5. We also qualitatively compared the topic words of the proposed method with those of other methods that do not apply de-biasing (mid and bottom of Table 11). While the others extract platform-specific topic words, PITopic reflects a more unbiased set.

To further verify if the generated topic words are less dependent on platforms and semantically describe the topics, we conducted an additional experiment with the ChatGPT API. We input only the topic words to the API and prompted it to name the topics accordingly (see Table 10 in Appendix). From the result, we could conclude that the topics labeled by ChatGPT and the topic words are reasonably well-matched. A more detailed description can be found in Appendix A.5.

Cluster visualization. To understand to what extent platforms are invariant when modeling topics, we visualize the cluster assignments on the synthetic data over four models as depicted in Figure 3. PITopic demonstrated a more dispersed distribution of platforms (represented by colors), with documents (denoted as dots) being closely clustered, in contrast to LDA, which exhibited less defined clustering. Meanwhile, BERTopic revealed a tendency for documents from specific platforms to cluster around particular topics, and CTM showed most documents flocking to a few topic clusters.

6 Conclusion

This paper investigates the influence of platform bias on existing topic modeling algorithms by collecting text corpora from multiple platforms sharing the same keywords, "ChatGPT." To address the platform bias, we introduced PITopic, which incorporates platform-invariant contrastive learning and a novel jargon-filtering process.

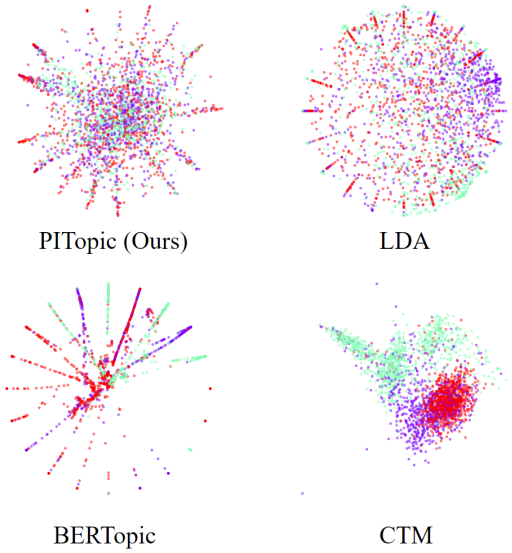

Figure 3: Clustering results over four topic models. Colors represent the three ground-truth class names (i.e., 20NewsGroup, BBC, and The New York Times), and dots depict the document cluster assignments.

As a result, our model extracted high-quality topics that are coherent across all platforms while being less affected by platform-specific biases, demonstrating superior topic coherence and diversity compared to other baselines. We believe our work enables more accurate topic discovery without bias, contributing to a variety of social analyses.

Limitations

This work can have a limitation due to the absence of a clear rule of thumb for setting the number of jargon, as the amount of jargon can vary across platforms. However, according to the hyper-parameter analysis in Appendix A.4, our model consistently outperforms baselines with varying hyper-parameters, indicating that selecting the optimal hyper-parameter is not challenging. Additionally, like other large language model (LLM)-based topic modeling methodologies, our approach can be influenced by the performance of the chosen backbone, as it remains fixed. Moving forward, we aim to develop methods for adaptively identifying jargon and fine-tuning the backbone itself to mitigate platform bias, addressing these limitations.

Furthermore, the current work only provides one case study with the keyword "ChatGPT" for real-world data, so it may be hard to claim if our model's performance is consistent across various cases. We plan to iterate the same experiments with other keywords from different domains to secure the generalization.

Some might consider the scope of the real use cases we addressed to be limited. In our work, we used the term "platform" to refer to the broader concept of a "source" where users' posts or articles are published. Our source-invariant topic modeling can be applied in various use cases. Although our experiments aggregated posts from multiple social media platforms, even a single platform (e.g., Reddit) contains distinct sources (e.g., threads) that can benefit from bias reduction when analyzing the overall discourse.

Ethics Statement

Although our work can potentially mitigate platform bias, we recognize that biases stemming from the pre-trained knowledge within the LLM backbone could persist in the topic modeling process. Furthermore, since our model aims to identify topics that are "independent" of the platforms, it may not fully capture opinions within individual platforms. When applying our model to high-stakes real-world applications, considering these factors, it is crucial to proceed with a thorough analysis.

Acknowledgement

This research was supported by a grant from the Korea Health Technology R&D Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health & Welfare, Republic of Korea (Grant number: HI19C1234). Also, this research was supported by the National Institute of Health (NIH) research project (Project No. 2024ER080300).

References

Nikolaos Aletras and Mark Stevenson. 2013. Evaluating topic coherence using distributional semantics. In *proc. of the International Conference on Computational Semantics (IWCS)*, pages 13–22.

Tumanov Alexander. 2023. New york times articles 1920-2020. Kaggle. Kaggle.com/datasets/tumanovalexander/nyt-articles-data.

Dimo Angelov. 2020. Top2vec: Distributed representations of topics. *arXiv preprint arXiv:2008.09470*.

Federico Bianchi, Silvia Terragni, and Dirk Hovy. 2020. Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. *arXiv preprint arXiv:2004.03974*.

Federico Bianchi, Silvia Terragni, Dirk Hovy, Debora Nozza, and Elisabetta Fersini. 2021. Cross-lingual contextualized topic models with zero-shot learning. In *proc. of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 1676–1683.

David M Blei, Thomas L Griffiths, and Michael I Jordan. 2010. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the ACM (JACM)*, 57(2):1–30.

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

Youngchul Cha and Junghoo Cho. 2012. Social-network analysis using topic models. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 565–574.

Ting Chen and Lala Li. 2020. Intriguing properties of contrastive losses. *arXiv preprint arXiv:2011.02803*.

Dongho Choi, Ziad Matni, and Chirag Shah. 2016. What social media data should i use in my research?: A comparative analysis of twitter, youtube, reddit, and the new york times comments. *Proceedings of the Association for Information Science and Technology*, 53(1):1–6.

Stephan A Curiskis, Barry Drake, Thomas R Osborn, and Paul J Kennedy. 2020. An evaluation of document clustering and topic modelling in two online social networks: Twitter and reddit. *Information Processing & Management*, 57(2):102034.

Cristian Danescu-Niculescu-Mizil, Robert West, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. No country for old members: User lifecycle and linguistic change in online communities. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 307–318.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *proc. of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4171–4186.

Adji B Dieng, Francisco J R Ruiz, and David M Blei. 2019. Topic modeling in embedding spaces. *arXiv preprint arXiv:1907.04907*.

Dan Fisher, Mark Kozdoba, and Shie Mannor. 2020. Topic modeling via full dependence mixtures. In *proc. of the International Conference on Machine Learning (ICML)*, pages 3188–3198.

Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. 2013. Using topic models for twitter hashtag recommendation. In *Proceedings of the 22nd international conference on world wide web*, pages 593–596.

Derek Greene and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd International Conference on Machine learning (ICML'06)*, pages 377–384. ACM Press.

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

Sungwon Han, Mingi Shin, Sungkyu Park, Changwook Jung, and Meeyoung Cha. 2023. Unified neural topic model via contrastive learning and term weighting. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1802–1817.

Thomas Hoffman. 1999. Probabilistic latent semantic analysis. In *proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, Roland Badeau, and Gustavo Rohde. 2019. Generalized sliced wasserstein distances. *Advances in Neural Information Processing Systems*, 32:261–272.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.

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

Martin Q Ma, Yao-Hung Hubert Tsai, Paul Pu Liang, Han Zhao, Kun Zhang, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2021. Conditional contrastive learning for improving fairness in self-supervised learning. *arXiv preprint arXiv:2106.02866*.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *proc. of the International Conference on Machine Learning*, pages 1727–1736.

Tom Mitchell. 1999. Twenty Newsgroups. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5C323.

Feng Nan, Ran Ding, Ramesh Nallapati, and Bing Xiang. 2019. Topic modeling with wasserstein autoencoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6345–6381.

David Newman, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. 2010. Evaluating topic models for digital libraries. In *Proceedings of*

the 10th annual joint conference on Digital libraries, pages 215–224.

Kunwoo Park, Haewoon Kwak, Jisun An, and Sanjay Chawla. 2021a. How-to present news on social media: A causal analysis of editing news headlines for boosting user engagement. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 15, pages 491–502.

Sungkyu Park, Sungwon Han, Jeongwook Kim, Mir Majid Molaie, Hoang Dieu Vu, Karandeep Singh, Jiyoung Han, Wonjae Lee, and Meeyoung Cha. 2021b. Covid-19 discourse on twitter in four asian countries: Case study of risk communication. *Journal of Medical Internet Research*, 23(3):e23272.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *proc. of the International Conference on Machine Learning (ICML)*, volume 242, pages 29–48.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3982–3992.

Suzanna Sia, Ayush Dalmia, and Sabrina J Mielke. 2020. Tired of topic models? clusters of pretrained word embeddings make for fast and good topics too! In *proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1728–1736.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. In *5th International Conference on Learning Representations (ICLR)*.

Viriya Taecharungroj. 2023. "what can chatgpt do?" analyzing early reactions to the innovative ai chatbot on twitter. *Big Data and Cognitive Computing*, 7(1):35.

Laure Thompson and David Mimno. 2020. Topic modeling with contextualized word representation clusters. *arXiv preprint arXiv:2010.12626*.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *proc. of the International Conference on Machine Learning (ICML)*, pages 9929–9939.

Guangxu Xun, Yaliang Li, Wayne Xin Zhao, Jing Gao, and Aidong Zhang. 2017. A correlated topic model using word embeddings. In *IJCAI*, volume 17, pages 4207–4213.

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. A biterm topic model for short texts. In *proc. of the Web Conference (WWW)*.

He Zhao, Lan Du, Wray Buntine, and Gang Liu. 2017. Metalda: A topic model that efficiently incorporates meta information. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 635–644. IEEE.

A Appendix

A.1 Code and Data Release

We are planning to release our code and data collected for evaluation upon acceptance.

A.2 Equation for MI between Platforms and Topics

The mutual information (MI) between two random variables P (Platforms) and T (Topics) quantifies the extent of information shared between these variables. It is represented by the following formula: $I(P;T)$ (Eq. 1). MI can be decomposed as the following equation:

$$
\begin{aligned}
I(P;T) &= \sum_{p}\sum_{t} p(p,t)\log\frac{p(p,t)}{p(p)p(t)} \\
&= \sum_{p}\sum_{t} p(p,t)\left(\log p(p,t) - \log p(p) - \log p(t)\right) \\
&= \sum_{p}\sum_{t} p(p,t)\log p(p,t) - \sum_{p}\sum_{t} p(p,t)\log p(p) \\
&\quad - \sum_{p}\sum_{t} p(p,t)\log p(t) \\
&= -H(P,T) + H(P) + H(T) \\
&= -H(T|P) + H(T)
\end{aligned}
$$

A.3 Implementation Details of PITopic

We employed the pre-trained language model Sentence-BERT, specifically the allMiniLM-L6-v2 variant (Reimers and Gurevych, 2019), for our analysis. A three-layer MLP is appended on top of the language model's [CLS] token embedding to extract the topic embedding $\mathbf{z}$. Then, a one-layer perceptron model without bias is applied to reconstruct the BoW representation. The model was trained over 100 epochs, selecting the model of the best epoch with the highest coherence score when regarding the validation set as a reference corpus. Data is split into train/test with a 7:3 ratio, where 10% of the train data was set aside for validation. The training process was optimized using the Adam optimizer, with an initial learning rate of 2e-2 and a batch size of 32. To prevent overfitting and ensure model stability, dropout, and batch normalization were employed, with the dropout set at 0.2. The Dirichlet prior for $\mathcal{T}_{\text{prior}}$ was set to 0.05 (i.e., 1/# of topics). The number of topics K was set to 20 and the size of the BoW vocabulary was set to 30,000. The number of jargon filtered N was set to 100 per platform (i.e., a total of 300). When performing contrastive learning, the nearest neighbor of a given sample is discovered using cosine similarity between text representations that are extracted from the language model via mean pooling.

The computing resources used for running our model consisted of a single A6000 GPU, with the training process completed within an hour.

A.4 Hyper-parameter Analysis

In this subsection, we analyze the effect of the number of jargon filtered. Specifically, for each platform, we vary the jargon count filtered by 300, 600, and 900 (i.e., 100, 200, and 300 per each platform). According to the results in Table 5, we confirm that the number of words did not substantially influence the topic coherence derived by the model.

Total number of jargon filtered	Mutual Information	Topic diversity	NPMI X	NPMI Reddit	NPMI YouTube	NPMI Average	UCI X	UCI Reddit	UCI YouTube	UCI Average
300 (100 per platform)	0.028	0.955	0.526	0.767	0.644	0.646	4.529	3.551	5.155	4.412
600 (200 per platform)	0.031	0.940	0.336	0.721	0.488	0.515	2.816	3.904	4.049	3.590
900 (300 per platform)	0.035	0.970	0.523	0.630	0.477	0.543	4.293	3.672	3.917	3.961

Table 5: The effects of the number of jargon filtered.

Model	Mutual Information	Topic Diversity	NPMI				UCI			
			X	Reddit	YouTube	Average	X	Reddit	YouTube	Average
LDA (10)	0.083	0.680	0.644	0.230	0.128	0.334	0.408	0.966	0.740	0.705
LDA (20)	0.100	0.705	0.255	0.315	0.299	0.290	2.027	2.170	2.331	2.176
LDA (30)	0.107	0.700	0.077	0.201	0.136	0.138	0.519	1.090	0.873	0.827
Total LDA	0.097	0.695	0.325	0.249	0.188	0.254	0.985	1.409	1.315	1.236
AVITM (10)	0.002	0.630	0.088	0.289	0.133	0.170	0.621	0.973	0.782	0.792
AVITM (20)	0.010	0.390	0.070	0.192	0.110	0.124	0.363	0.887	0.582	0.611
AVITM (30)	0.006	0.336	0.081	0.261	0.116	0.153	0.488	1.082	0.664	0.745
Total AVITM	**0.006**	0.452	0.080	0.247	0.120	0.149	0.491	0.981	0.676	0.716
Top2Vec (10)	0.490	0.580	0.128	0.265	0.234	0.209	0.992	1.918	1.798	1.569
Top2Vec (20)	0.495	0.535	0.153	0.263	0.233	0.216	1.236	1.980	1.820	1.679
Top2Vec (30)	0.486	0.586	0.170	0.264	0.250	0.228	1.372	1.976	1.944	1.764
Total Top2Vec	0.490	0.567	0.150	0.264	0.239	0.218	1.200	1.958	1.854	1.671
ClusterTM (10)	0.078	0.630	0.106	0.238	0.166	0.170	0.744	0.953	0.991	0.896
ClusterTM (20)	0.088	0.465	0.101	0.238	0.149	0.163	0.704	0.946	0.905	0.852
ClusterTM (30)	0.091	0.476	0.093	0.203	0.152	0.149	0.634	1.044	0.901	0.860
Total ClusterTM	0.086	0.524	0.100	0.226	0.156	0.161	0.694	0.981	0.932	0.869
CTM (10)	0.183	0.850	0.143	0.352	0.178	0.224	1.097	1.770	1.268	1.378
CTM (20)	0.176	0.855	0.212	0.363	0.216	0.264	1.511	2.114	1.578	2.034
CTM (30)	0.202	0.813	0.245	0.346	0.216	0.269	1.812	2.067	1.594	1.824
Total CTM	0.187	0.839	0.200	0.354	0.203	0.252	1.473	1.984	1.480	1.746
BERTopic (10)	0.039	0.980	0.387	0.421	0.359	0.389	3.140	3.060	2.768	2.989
BERTopic (20)	0.085	0.950	0.255	0.316	0.299	0.290	2.027	2.170	2.331	2.176
BERTopic (30)	0.096	0.940	0.324	0.353	0.333	0.337	2.669	2.600	2.686	2.652
Total BERTopic	0.073	0.957	0.322	0.363	0.330	0.339	2.612	2.610	2.595	2.606
UTopic (10)	0.278	0.790	0.095	0.215	0.126	0.145	0.686	1.055	0.840	0.860
UTopic (20)	0.311	0.610	0.092	0.200	0.132	0.301	0.642	1.105	0.862	0.870
UTopic (30)	0.309	0.510	0.104	0.209	0.130	0.148	0.750	1.159	0.862	0.924
Total UTopic	0.299	0.637	0.097	0.208	0.129	0.198	0.693	1.106	0.855	0.885
PITopic (10)	0.024	1.000	0.387	0.821	0.290	0.499	3.370	4.145	2.527	3.347
PITopic (20)	0.028	0.955	0.526	0.767	0.644	0.646	4.529	3.551	5.155	4.412
PITopic (30)	0.044	0.963	0.438	0.464	0.426	0.443	2.814	2.741	3.195	2.917
Total PITopic (Ours)	0.032	**0.973**	**0.450**	**0.684**	**0.453**	**0.529**	**3.571**	**3.479**	**3.626**	**3.559**

Table 6: Performance evaluation over real-world data. Results are averaged across different numbers of topics - 10, 20, and 30. The best performances are highlighted.

Regarding the number of topics as a hyper-parameter, before conducting topic modeling, we performed experiments analyzing the distribution of topics using LDA and GPT-3.5 with the various numbers. In consequence, 20 showed the best topic coherence, so we reported outcomes mainly with the 20 numbers of topics on our subsequent experiments in the main body of the paper. The entire set of experimental results altering the number of topics to 10, 20, and 30 is listed in Table 6,7; we could confirm that the trends did not significantly change across the numbers.

To comprehensively understand the sole impact of our proposed model structure (Figure 2) on topic modeling, we also applied the jargon filtering process to other models (LDA, CTM, and BERTopic). Table 8 shows that PITopic outperforms the other models that injected jargon-filtered data across all evaluation metrics. Meanwhile, the jargon process can boost the performance margin as shown in Table 4, so to sum up, these findings suggest effectively integrating jargon filtering and model structure is important to mitigate platform-specific biases in topic modeling.

Model	Mutual Information	Topic Diversity	NPMI				UCI			
			X	Reddit	YouTube	Average	X	Reddit	YouTube	Average
LDA (10)	0.386	0.670	0.252	0.152	0.431	0.278	1.002	0.422	2.258	1.227
LDA (20)	0.400	0.555	0.164	0.427	0.268	0.286	0.530	2.222	1.082	1.278
LDA (30)	0.426	0.553	0.279	0.161	0.440	0.293	1.168	0.547	2.287	1.334
Total LDA	0.404	0.593	0.232	0.247	0.380	0.286	0.900	1.064	1.876	1.280
AVITM (10)	0.020	0.750	0.269	0.073	0.521	0.288	1.293	0.262	2.722	1.426
AVITM (20)	0.011	0.690	0.138	0.585	0.289	0.337	0.454	3.118	1.446	1.673
AVITM (30)	0.023	0.667	0.295	0.099	0.515	0.303	1.471	0.352	2.743	1.522
Total AVITM	**0.018**	0.702	0.234	0.252	0.442	0.309	1.073	1.244	2.304	1.540
Top2Vec (10)	0.888	0.790	0.556	0.283	0.637	0.492	2.776	1.117	3.493	2.462
Top2Vec (20)	0.860	0.530	0.298	0.581	0.548	0.476	1.291	3.139	2.848	2.426
Top2Vec (30)	0.894	0.591	0.475	0.287	0.689	0.484	2.472	1.187	3.748	2.469
Total Top2Vec	0.881	0.637	0.443	0.384	0.625	0.484	2.180	1.814	3.363	2.452
ClusterTM (10)	0.342	0.870	0.334	0.208	0.464	0.335	1.639	0.704	2.387	1.577
ClusterTM (20)	0.500	0.465	0.218	0.505	0.342	0.355	0.795	2.656	1.669	1.707
ClusterTM (30)	0.544	0.770	0.319	0.213	0.489	0.340	1.541	0.780	2.583	1.635
Total ClusterTM	0.462	0.702	0.290	0.309	0.432	0.344	1.325	1.380	2.213	1.639
CTM (10)	0.308	0.890	0.465	0.356	0.575	0.465	2.434	1.355	3.076	2.288
CTM (20)	0.353	0.820	0.392	0.302	0.596	0.430	2.017	1.239	3.191	2.149
CTM (30)	0.456	0.733	0.410	0.235	0.619	0.421	2.087	0.737	3.355	2.060
Total CTM	0.372	0.814	0.422	0.298	0.597	0.439	2.179	1.110	3.207	2.166
BERTopic (10)	0.522	0.930	0.407	0.250	0.486	0.381	2.032	1.037	2.581	1.883
BERTopic (20)	0.565	0.900	0.230	0.532	0.394	0.385	0.969	2.841	1.956	1.922
BERTopic (30)	0.567	0.880	0.400	0.278	0.565	0.414	2.020	1.216	3.000	2.079
Total BERTopic	0.551	0.903	0.346	0.353	0.482	0.393	1.674	1.698	2.512	1.961
UTopic (10)	0.549	0.830	0.400	0.227	0.450	0.359	1.947	0.854	2.29	1.697
UTopic (20)	0.676	0.830	0.244	0.533	0.430	0.402	0.968	2.789	2.091	1.949
UTopic (30)	0.693	0.753	0.404	0.238	0.489	0.377	2.050	0.896	2.600	1.849
Total UTopic	0.639	0.804	0.349	0.332	0.456	0.379	1.655	1.513	2.327	1.832
PITopic (10)	0.028	0.980	0.715	0.397	0.594	0.569	3.573	1.840	2.915	2.776
PITopic (20)	0.051	0.985	0.352	0.790	0.584	0.575	1.871	4.216	3.176	3.088
PITopic (30)	0.077	0.980	0.432	0.200	0.790	0.474	2.211	0.899	4.154	2.421
Total PITopic (Ours)	0.052	**0.982**	**0.500**	**0.462**	**0.656**	**0.539**	**2.552**	**2.318**	**3.415**	**2.762**

Table 7: Performance evaluation over synthetic data. Results are averaged across different numbers of topics - 10, 20, and 30. The best performances are highlighted.

Model	Mutual Information	Topic Diversity	NPMI				UCI			
			X	Reddit	YouTube	Average	X	Reddit	YouTube	Average
LDA	0.076	0.73	0.088	0.219	0.134	0.147	0.609	1.001	0.826	0.812
CTM	0.171	0.855	0.168	0.375	0.273	0.272	1.278	2.312	2.105	1.898
BERTopic	0.091	0.955	0.313	0.366	0.307	0.329	2.54	2.596	2.386	2.507
PITopic (Ours)	**0.028**	**0.955**	**0.526**	**0.767**	**0.644**	**0.646**	**4.529**	**3.551**	**5.155**	**4.412**

Table 8: Evaluation metrics on real-world data with jargon filtering. The number of topics was set to 20. The best performances are highlighted.

	Total jargon list
X	coinex, announces, seos, chatgptpowered, launches, bigdata, unveils, openaichatgpt, tags, hn, stablediffusion, chatgptstyle, reportedly, mba, marketers, baidu, technews, fintech, chatgptlike, elonmusk, notion, goog, googleai, digitalmarketing, artificalintelligence, rt, googl, bardai, edtech, malware, wharton, agix, chatgptplus, datascience, deeplearning, msft, weirdness, tweets, amid, aitools, cybersecurity, airdrop, cc, valentines, startups, snapchat, generativeai, buzzfeed, fastestgrowing, anthropic, maker, rival, techcrunch, aiart, nocode, invests, cybercriminals, abstracts, nyc, webinar, retweet, educators, brilliance, rescue, daysofcode, gm, rtechnology, linkedin, licensing, copywriting, copywriters, contentmarketing, revolutionizing, technologynews, warns, metaverse, cofounder, trending, founders, aipowered, openaichat, releases, microsofts, chinas, infosec, launching, jasper, nfts, newsletter, chatgptgod, futureofwork, digitaltransformation, founder, feb, buzz, rn, ux, courtesy, nick, claude
Reddit	remindme, giphy, gif, deleted, giphydownsized, chadgpt, removed, patched, nerfed, yup, waitlist, refresh, sydney, mods, nsfw, characterai, screenshot, downvoted, youcom, meth, ascii, karma, hahaha, hangman, chatopenaicom, emojis, porn, redditor, vpn, upvotes, blah, upvote, violated, yep, joking, nope, offended, mod, bruh, roleplay, ops, bob, dans, redditors, nerf, firefox, trolling, sarcastic, huh, turbo, troll, patch, tag, url, sus, erotica, chad, gotcha, basilisk, login, lmfao, temperature, poll, emoji, rick, dm, jailbreak, orange, sub, quack, davinci, uh, flagged, op, markdown, flair, cares, refreshing, hitler, cookies, hmm, yikes, erotic, gti, paywall, elaborate, yea, ah, uncensored, rude, colour, bitch, therapy, neutered, deny, chats, jailbroken, cake, dungeon, dang
YouTube	zronx, tuce, jontron, levy, bishop, rook, thumbnail, quotquot, jon, linus, hrefaboutinvalidzcsafeza, beluga, vid, bhai, gemx, raid, ohio, circle, subscribed, anna, stare, canva, napster, shapiro, sponsor, broker, websiteapp, manoj, subscriber, bluewillow, alex, vids, legends, ryan, shes, hackbanzer, quotoquot, pictory, youtuber, profitable, pawn, joma, folders, lifechanging, thomas, ur, plz, mike, scott, casey, adrian, enjoyed, stockfish, invideo, shortlisted, hikaru, bless, corpsb, chatgbt, bfuture, curve, accent, amc, tutorials, gotham, mrs, earning, bra, elo, oliver, youtubers, quotcontinuequot, membership, labels, dagogo, eonr, hai, quotai, affiliate, congratulationsbryou, subscribers, thumbnails, azn, beast, tom, trader, garetz, quot, subbed, pls, quotchatgpt, gtp, machina, quoti, bret, terminator, watchingbrdm, quothow, nowi, mint

Table 9: List of words with top-100 jargon scores computed by our proposed method for each platform.

A.5 Extra Qualitative Analyses

We showcased the comprehensive list of jargon identified using the c-TF-IDF technique, aggregating 300 words by selecting 100 words specific to each platform (Table 9). This list highlights words predominantly utilized on each platform, such as 'rt', 'retweet', 'gif', and 'removed' for Reddit, along with 'subscribed' for YouTube, demonstrating the platform-specific usage patterns captured by our model PITopic.

Additionally, We extracted topics from the data containing the keyword "ChatGPT" across the three platforms (X, Reddit, and YouTube). Table 11 (top) reports the top 10 topic words extracted by PITopic when trained on real-world data. We found that the topic words related to the target keyword, ChatGPT, have been steadily extracted within each topic. For instance, we confirm that Topic #1, for example, deals with technological advances and explainable AI. The words "xai" and "neural" composing this topic implicitly refer to the technological advances that ChatGPT implements and how the language model makes decisions. Topic #5 explores the usefulness of ChatGPT and how it differs from other AI tools. It includes a comparison of ChatGPT with tools such as Google Assistant and user feedback such as "fantastic" and "supportive." The word "textdavinci" in topic #11 refers to OpenAI's other LLM, which offers similar features to ChatGPT and includes a comparison between the two models and a discussion of use cases. Topic #13 is to create stories while #14 discusses educational use and creative writing and provides examples of the use of ChatGPT for educational purposes, as indicated by "teachergpt" and "printer."

One remark is that the platforms we used focus on user-centered content where individuals can freely express their opinions and share information. Due to the nature of the data collected from these sources, the raw, unfiltered language of online communities is reflected in the top 10 words. Nevertheless, the top 10 words show that users of the platforms discuss a wide range of topics, including the use and application of ChatGPT.

We also qualitatively assess the topic words of the other comparable models LDA and CTM, which may be affected by platform bias; note that only models with computable word distributions per topic based on pLSA assumption were subject to be verified. The extracted topic words for each model can be found in Table 11 (mid and bottom). We observe that 1) in the case of LDA, specific topic words are redundantly extracted (e.g., "ChatGPT", "AI"), indicating the model's inability to generate unique topics, and 2) in the case of CTM, we can see that the topics are influenced by the platforms, especially with Topic #12, where the topic words represent the characteristic of the "YouTube" platform.

Next, to check whether the extracted topic words represent the semantically fitted topics we further experimented; the top 20 topic words per topic extracted from PITopic were entered into the ChatGPT API and asked to name each topic the corresponding group of the topic words conveys. The relationship between the topic words extracted from PITopic and the associated topics was qualitatively evaluated as shown in Tables 10 and 11 (top). The result validates that our model is capable of extracting topics that are sufficiently coherent, and independent of the platform.

Topic	ChatGPT-summarized topics from topic words
0	Internet culture and meme
1	Technical advance and explainable AI
2	Negative feedback and reactions
3	Cryptocurrency and economic discussions
4	Interaction and communication between users
5	ChatGPT versatility and comparison
6	Cultural issues and representation
7	Life and stories
8	Community management and bot usage
9	Writing and content creation
10	Food and humor
11	Manage online communities and bots
12	Political issues and debates
13	Writing and creating
14	Education and creative writing
15	Friendships and social media
16	Empathy and robotizationr
17	Management and operational issues
18	AI and social impact
19	Share daily life and experiences

Table 10: Summarized topic labels using the ChatGPT API based on the top 20 topic words per topic, derived by PITopic for the keyword "ChatGPT" in the real-world data.

Topic	Top-10 PITopic topic words in real-world data
0	whore, hilarious, wwwboyfriendgptcom, clippygpt, vanoss, copypasta, chilling, rofl, tshirt, shatgpt
1	neural, explainable, redefinition, calculation, xai, soso, operates, relearn, mathematical, kdnuggets
2	inferred, burp, sucker, blaming, cringe, dumbass, cocaine, pete, hahahah, erit
3	uganda, currency, dollar, fiat, lottery, coingecko, sellside, bitcoin, procrastinating, aitoken
4	messaged, stepchange, oof, youfrom, exemplar, danial, copyrighted, kakkar, quotultimate, toolbarquot
5	agreed, appoint, chatgptgtgoogleassistant, preach, fantastic, supportive, maximizing, sonu, apologetic, protip
6	emojies, antihindu, toying, fu, probaron, amp, ftw, oof, quetta, mem
7	yus, uuugggg, spaghetti, egypt, vunnai, elanti, storytho, pyramid, dong, ruuullleeesss
8	discord, performed, moderator, prevent, friendly, compose, subreddit, experiment, bot, contact
9	copypasta, portugalcykablyat, thesaurus, verbalize, refrence, proceeds, pivoting, brazil, contribution, upvoted
10	tomato, brutal, tinkled, cherredith, cherry, pineapple, jelly, franz, soda, lmao
11	server, moderator, comment, bot, textdavinci, contact, discord, friendly, compose, performed
12	radical, leftwing, supremists, fascist, leftist, anticompetitive, stagnation, vat, lunatic, boycott
13	scripters, storywork, screenplay, yesteryear, detective, robs, writingcommmunity, comedy, conciseness, lagaan
14	teachergpt, printer, filthy, doggerel, poet, genuary, thurber, paperclip, thermonuclear, chatgptwritten
15	bff, serach, pawer, kaha, download, bud, wesh, dogg, snoop, puro
16	empathy, robotized, worldbrthe, writersofinstagram, writerscommunity, cutest, datadriveninvestor, privy, contractpilled, screwdriver
17	cleaning, shehmmm, signups, deleting, exploiting, omfg, communist, wetting, compromising, outsourcing
18	artificialintelligence, evolveit, disrupt, thedigitalexecutive, inept, sociobits, trolly, gtthat, intensifying, journalist
19	texted, queue, samaritan, slept, wellshit, exhales, dishwasher, username, failed, broke
Topic	**Top-10 LDA topic words in real-world data**
0	code, chatgpt, just, text, doe, write, use, number, work, language
1	bot, comment, prompt, discord, chatgpt, server, yes, r, gpt, actual
2	gpt, chat, like, people, just, think, thing, know, ai, say
3	video, chatgpt, like, just, good, know, really, make, use, thing
4	time, chatgpt, money, market, just, said, pay, people, trading, make
5	chatgpt, book, friend, year, assistant, left, way, ai, new, knew
6	chatgpt, prevent, search, bing, microsoft, developer, openai, engine, mode, app
7	gt, data, information, god, thank, training, chatgpt, trump, knowledge, pattern
8	ai, human, job, people, going, think, need, work, machine, make
9	lol, just, chatgpt, man, shit, oh, told, did, know, wow
10	chatgpt, intelligence, artificial, news, music, song, love, elon, mr, option
11	chatgpt, amp, day, phone, great, using, awesome, ai, fun, like
12	subreddit, like, people, real, just, computer, problem, ai, think, program
13	ai, chatgpt, model, content, language, response, text, prompt, generate, information
14	ai, chatgpt, robot, world, story, like, joke, humanity, future, mind
15	chatgpt, used, moderator, message, concern, user, reply, automatically, multiple, free
16	repetitive, game, dan, textdavinci, character, chatgpt, play, rule, apply, provide
17	google, chatgpt, ai, chatbot, technology, internet, company, openai, like, open
18	chatgpt, answer, question, ask, write, asked, work, writing, student, just
19	request, compose, word, thanks, chatgpt, thread, monkey, guess, letter, apple
Topic	**Top-10 CTM topic words in real-world data**
0	unlike, france, clever, picnic, stick, disappointed, allies, branches, tay, immersion
1	ai, could, would, think, like, make, data, also, human, time
2	vegetable, ensuring, assess, problemsolving, demanding, satisfaction, speculate, association, originality, outrageous
3	notion, tags, android, database, bings, launches, rival, via, chatgptpowered, saved
4	lmao, bro, king, intro, gradually, anna, ex, thumbs, picnic, miles
5	api, search, copy, extension, website, app, create, chrome, blog, prompts
6	order, everyone, prevent, request, users, reply, free, well, multiple, yes
7	people, think, human, humans, life, like, even, dont, already, things
8	code, answer, answers, writing, write, ask, question, words, give, correct
9	doesnt, isnt, dont, probably, cant, wont, enough, stuff, wrong, students
10	picnic, alice, facial, firstly, vegetable, outrageous, chosen, internalized, deem, impress
11	artificialintelligence, tech, chatbot, artificial, machinelearning, openai, microsoft, news, generative, dalle
12	sir, thanks, thank, video, videos, youtube, awesome, sharing, glad, watching
13	users, prevent, order, reply, well, request, free, yes, multiple, public
14	picnic, randomly, chosen, presenting, defined, symbols, rant, texture, musical, fruits
15	trading, market, pay, month, days, months, paid, profit, year, last
16	joke, robot, shit, funny, poem, asked, told, movie, god, fake
17	bot, chat, prompt, gpt, bing, comment, chatgpt, models, try, public
18	anything, like, something, know, would, make, one, also, answer, cannot
19	excessive, pulling, outrageous, clinton, picnic, trustworthiness, verifiable, mandatory, firstly, speculate

Table 11: Top-10 topic words discovered from each model (PITopic, LDA, and CTM) for the keyword "ChatGPT" in the real-world data.