Federated Topic Modeling

Di Jiang
AI Group, WeBank Co., Ltd
Shenzhen, China
dijiang@webank.com

Yuanfeng Song
AI Group, WeBank Co., Ltd
Shenzhen, China
yfsong@webank.com

Yongxin Tong
BDBC, SKLSDE Lab and IRI
Beihang University
Beijing, China
yxtong@buaa.edu.cn

Xueyang Wu
The Hong Kong University of Science and Technology
Hong Kong SAR, China
xwuba@cse.ust.hk

Weiwei Zhao
AI Group, WeBank Co., Ltd
Shenzhen, China
davezhao@webank.com

Qian Xu
AI Group, WeBank Co., Ltd
Shenzhen, China
qianxu@webank.com

Qiang Yang
AI Group, WeBank Co., Ltd
Shenzhen, China
qiangyang@webank.com

ABSTRACT

Topic modeling has been widely applied in a variety of industrial applications. Training a high-quality model usually requires massive amount of in-domain data, in order to provide comprehensive co-occurrence information for the model to learn. However, industrial data such as medical or financial records are often proprietary or sensitive, which precludes uploading to data centers. Hence training topic models in industrial scenarios using conventional approaches faces a dilemma: a party (i.e., a company or institute) has to either tolerate data scarcity or sacrifice data privacy. In this paper, we propose a novel framework named Federated Topic Modeling (FTM), in which multiple parties collaboratively train a high-quality topic model by simultaneously alleviating data scarcity and maintaining immune to privacy adversaries. FTM is inspired by federated learning and consists of novel techniques such as private Metropolis Hastings, topic-wise normalization and heterogeneous model integration. We conduct a series of quantitative evaluations to verify the effectiveness of FTM and deploy FTM in an Automatic Speech Recognition (ASR) system to demonstrate its utility in real-life applications. Experimental results verify FTM’s superiority over conventional topic modeling.

CCS CONCEPTS

• Information systems → Data mining. • Computing methodologies → Topic modeling.

KEYWORDS

Text Semantics; Topic Model; Bayesian Networks

ACM Reference Format:

1 INTRODUCTION

Topic modeling has been successfully used in many industrial applications, from military analysis [41], to web search log mining [8, 23, 24], to medical informatics [1, 26, 36]. As training a high-quality topic model for a specific application typically requires comprehensive in-domain data to provide sufficient co-occurrence information, relying on data collected from a single party will be faced with the challenge of data scarcity. Meanwhile, since these data are typically proprietary and sensitive, regulations such as the newly enforced European Union General Data Protection Regulation (GDPR) [7, 42, 43] may preclude uploading them to data centers and being utilized in a centralized approach. These two critical problems pose new challenges to conventional topic modeling, which we refer to as the state-of-the-art distributed architectures [35, 54, 55] for training topic models on computer clusters within a data center.

To solve the above problems, a new topic modeling paradigm simultaneously alleviating data scarcity and protecting data privacy is urgently needed in industry. However, the huge discrepancy between the scenario of conventional topic modeling and that studied in this paper results in three challenging research issues. First, how to protect the privacy of training data of each party from adversaries. Privacy is typically neglected in conventional topic modeling and anyone who is able to access computing nodes or monitor network communication can easily get a glimpse of the data of each party. Such practice is increasingly forbidden by new data regulations. Second, how to reduce the communication cost
between computing nodes. Conventional topic modeling such as those deployed upon MapReduce [55] or ParameterServer [54] usually has demanding requirement of communication efficiency that is only satisfied by a data-center-grade network. However, in the present problem, different parties may be located in different data centers and connected by low bandwidth. Hence, it is infeasible to allow computing nodes to frequently communicate with each as before. Third, how to handle the variety of data and models across different parties. Conventional topic modeling relies upon the assumption that different computing nodes store independent and identically distributed (i.i.d.) data and trains the same topic model. However, this requirement can hardly be met in the present problem where each party usually stores highly unbalanced data and trains heterogeneous topic models (i.e., topic models with different regularity).

Inspired by the concept of federated computation, which refers to a distributed architecture that a master coordinates a fleet of parties to compute aggregated statistics of private data [20, 33], we propose a new framework named Federated Topic Modeling (FTM) that solved the aforementioned problems in a principled approach. As shown in Figure 1, FTM is composed of two computational components: party computation and master computation. Party computation provides a flexible mechanism for balancing model utility and data privacy. It seamlessly integrates differential privacy with Markov Chain Monte Carlo (MCMC) [16] for both private and i.i.d. data and trains the same topic model. Notably, this requirement can hardly be met in the present problem where each party usually stores highly unbalanced data and trains heterogeneous topic models (i.e., topic models with different regularity).

The present work is related to a broad range of literature. We review the most related works in the fields of topic modeling, federated learning and differential privacy respectively.

## 2 RELATED WORK

The recent advancement of this field focuses on training topic models such as LDA on clusters by distributed computing [35, 51, 54, 55], and LightLDA [54] maintains the state-of-the-art performance in terms of model quality. We consider these works as conventional topic modeling and the issue of privacy is typically neglected in them. A recent work [39] proposes a technique to privatize the parameters of variational inference, however, this technique is based upon a single computing node and is not straightforward to apply in a distributed computing. Although privacy and distributed computing are two critical factors determining the applicability of topic models in industry, they have been studied independently in existing work and how to integrate them in a unified framework is still an open problem.

### 2.1 Topic Modeling

Topic modeling has been intensively studied and widely used for the last decade. Latent Dirichlet Allocation (LDA) [4] plays the most important role among all kinds of topic models. In the meantime, various extensions of LDA have been proposed in order to satisfy specific needs in different domains and applications. For example, Topic-over-time (TOT) [45] captures the changes of latent topic over times by jointly modeling the timestamps with word occurrence patterns. Supervised LDA [32] makes use of the labelled documents to guide the inference of the topics. Sentence LDA [2] incorporates the structure of the text during the model training and inference processes by restricting all the words in the same sentence sharing the same topic.

The recent advancement of this field focuses on training topic models such as LDA on clusters by distributed computing [35, 51, 54, 55], and LightLDA [54] maintains the state-of-the-art performance in terms of model quality. We consider these works as conventional topic modeling and the issue of privacy is typically neglected in them. A recent work [39] proposes a technique to privatize the parameters of variational inference, however, this technique is based upon a single computing node and is not straightforward to apply in a distributed computing. Although privacy and distributed computing are two critical factors determining the applicability of topic models in industry, they have been studied independently in existing work and how to integrate them in a unified framework is still an open problem.

### 2.2 Federated Learning

Many machine learning tasks have been achieving significant progress recently years with the development of machine learning techniques, such as deep neural networks, and the increasing amount of labeled data. However, high-quality labeled data are precious, which take huge human resources and time to collect and to annotate. Moreover, in many practical applications, the accessible data are very limited, making the learning difficult. To address the small data problem, some researchers focus on developing machine learning algorithms that can leverage the knowledge from a domain with rich data to help the learning in the target domain with limited data or even without labeled data, i.e., transfer learning [37]. The other proposed solution to address the small data problem is trying to join the fragmented small data from multi-party, i.e. collaborative machine learning. However, both of these solutions do not consider the data privacy and security, while personal information privacy and security have become an insurmountable concern before the wide applications of these techniques. Many countries enforce or plan to publish laws and regulations to protect the personal information privacy and security. For example, GDPR aims to protect the security and privacy of user, giving the right to user for ripping their personal information from companies.
In addition, due to the probabilistic property of differential privacy, researchers propose privacy-preserving collaborative learning paradigms, such as vertical federated learning [10] and federated transfer learning [5]. According to how the data across parties are utilized in federated learning, algorithms can be categorized into three types [50]:

1. **Horizontal federated learning**: data on different parties have the same feature space, and the samples on different parties are different. Samples are learned as virtually joint together across parties with protection of privacy and security.

2. **Vertical federated learning**: data on different parties share the same sample ID space, and their feature spaces are different. The data are virtually joint in feature dimension through vertical federated learning, i.e., the feature of samples in each party are extended with features from other parties.

3. **Federated transfer learning**: the sample ID spaces and features spaces of data in different parties are different. Samples are learned as virtually joint together across parties with protection of privacy and security.

To deal with the contradiction between the need of massive data for learning and the regulations to protect the data privacy and security, researchers propose privacy-preserving collaborative learning paradigms, such as neural networks [33], SVMs [6], Logistic regression [18]. According to how the data across parties are utilized in federated learning, algorithms can be categorized into three types [50]:

1. **Horizontal federated learning**: data on different parties have the same feature space, and the samples on different parties are different. Samples are learned as virtually joint together across parties with protection of privacy and security.

2. **Vertical federated learning**: data on different parties share the same sample ID space, and their feature spaces are different. The data are virtually joint in feature dimension through vertical federated learning, i.e., the feature of samples in each party are extended with features from other parties.

3. **Federated transfer learning**: the sample ID spaces and features spaces of data in different parties are different. Samples are learned as virtually joint together across parties with protection of privacy and security.

The typical scenario of horizontal federated learning is the learning applications on mobile devices, such as language modeling [21], and keyword spotting [28]. Data in mobile devices are highly personal and thus sensitive, while in federated learning, data are only operated locally, without transferring outside the party. To some extent, the privacy and security can be guaranteed by accessing the data locally. However, transferring parameters or gradients may also leak sensitive information [34, 44, 48]. To achieve higher privacy and security level, [5] improve the horizontal federated learning algorithm by introducing the secure aggregation protocol. In addition, due to the probabilistic property of differential privacy [11], differentially private federated learning is widely used in federated learning for protecting personal and thus sensitive, while in federated learning, data are only operated locally, without transferring outside the party. To some extent, the privacy and security can be guaranteed by accessing the data locally. However, transferring parameters or gradients may also leak sensitive information [34, 44, 48]. To achieve higher privacy and security level, [5] improve the horizontal federated learning algorithm by introducing the secure aggregation protocol. In addition, due to the probabilistic property of differential privacy [11], differentially private federated learning is widely used in federated learning for protecting the transaction of models or data [3, 15, 19, 22, 38, 46]. In vertical federated learning [10] and federated transfer learning [31], homomorphic encryption (HE) [40] are used, as HE does not inject noise to the transferred data. However, HE-based methods bring extra encryption computation costs and larger communication cost due to the ciphertext. Secure multi-party computation [52] methods are also used for collaborative learning, but the number of participants in this protocol are limited. In our work, we use horizontal federated learning techniques with differentially private protection on data privacy.

### 2.3 Differential Privacy

Differential privacy provides a quantitative definition on data and model that is compatible with many machine learning algorithms, which relieve the paradox of privacy-preserving learning that grapes helpful information from a population but learns nothing about any individual. The formal definition of differential privacy is as follows: A randomized algorithm $M(X)$ is considered as $(\epsilon, \delta)$-differentially private if

$$Pr(M(X \in S)) \leq e^\epsilon Pr(M(X' \in S)) + \delta$$

for all $S \subseteq \text{Range}(M)$, and for all adjacent datasets $X, X'$. Here differential privacy leaves the definition of adjacent open to different settings. It means that one can infer privacy information from the output of that randomized algorithm $M(X)$ with a negligible probability, where $X$ can be regarded as any subset from a population. One can sniff individual information through enough many queries to elaborately constructed populations. Therefore, the difference of output of an algorithm with different $X$ is critical to the individual privacy protection. In differential privacy, it is measured as sensitivity. Researchers develop many tools to guarantee a certain level of privacy under a given sensitivity, such as the Gaussian mechanism, and Laplace Mechanism. Laplace mechanism [12] is a common method for obtaining $\epsilon$-differential privacy. The main idea is to add Laplace noise to the revealed data with the amount of noise controlled by $\epsilon$. Specifically, the $L_1$ sensitivity $\Delta h$ for function $h$ is defined as:

$$\Delta h = \max_{X, X'} ||h(X) - h(X')||_1$$

**Figure 1**: The Federated Topic Modeling Framework: Party Computation (1) Private MCMC Sampling and 2) Topic-wise Normalization) and Master Computation (3) Integrating Local Topic Models and 4) Composing New Local Topic Models.)
for all datasets $X, X'$ differing in at most one element. The Laplace mechanism adds noise via:

$$M_L(X, h, e) = h(X) + (Y_1, Y_2, ..., Y_d),$$

$$Y_j \sim \text{Laplace}(\Delta h(e)), \forall j \in \{1, 2, ..., d\},$$

where $d$ is the dimensionality of $h$. The $M_L(X, h, e)$ mechanism is $e$-differentially private.

Composition theorems [12] deal with the case when we combine several differentially private blocks together for more sophisticated algorithms and differential privacy satisfies sequential composition and parallel composition. Sequential composition: suppose $M_i(X), j = 1, \cdots, l$, are $(\epsilon_j)$-differentially private, then the combination of these algorithms $X \rightarrow (M_1(X), \cdots, M_l(X))$ is $(\sum_j \epsilon_j)$-differentially private. In a special case, when all the $M_j$ are homogeneous, the combination yields $(\epsilon x, \epsilon y)$-differentially private. Parallel composition: let $X_i$ be arbitrary disjoint subsets of the input $X$, and $M_j(X_i), j = 1, \cdots, l$, are $(\epsilon_j)$-differentially private, then the combination of these algorithms $X \rightarrow (M_1(X_1) \cdots, M_l(X_l))$ satisfies $(\max_j \epsilon_j)$-differentially private. Besides composition of differentially private mechanisms, DP is also immune to any deterministic post-processing [13], which means we can take further operations over the privatized data, achieving the identical differential privacy budget.

## 3 FEDERATED TOPIC MODELING FRAMEWORK

The recent multitude of successful applications of topic modeling have almost relied on Latent Dirichlet Allocation (LDA) and variants of MCMC as the parameter inference algorithm [29, 54]. Thus, it is natural that we describe FTM by starting from LDA and MCMC. The techniques discussed in this section pave the way for designing similar algorithms for other topic models. In order to facilitate the discussion thereafter, we list the notations that will be used in this paper in Table 1. We first discuss the party computation in Section 3.1. Then we discuss how the master computation works in Section 3.2 and finally present the whole workflow of FTM in Section 3.3.

### 3.1 Party Computation

Each party is the workhorse for training its local topic model. We now discuss several novel mechanisms of party computation to protect the privacy of data stored on each party. We start with describing a private Gibbs sampling algorithm in Section 3.1.1 and then adapt it to a private Metropolis Hastings algorithm that enjoys much higher efficiency in Section 3.1.2. Finally, we discuss how to avoid revealing the original word distribution of the local model in Section 3.1.3.

#### 3.1.1 Private Gibbs Sampling

Gibbs sampling is widely used for parameter inference of topic models [17, 25, 45]. Let the words $w$ be the training data from a party and $z$ be the latent topics assigned to $w$. According to the generative assumption of LDA [4], the joint probability is as follows:

$$P(w, z, \Theta, \Phi | \alpha, \beta) = P(\Theta | \alpha) P(\Phi | \beta) P(z | \Theta) P(w | z, \Phi)$$

(4)

where $\Theta$ are the topic distributions of documents, $\Phi$ are the word distributions of topics, $\alpha$ and $\beta$ are hyperparameters that are usually fixed to constant values [17]. The Gibbs update of a topic $z_{di}$ that corresponds to the $i$th word $w$ in document $d$ is defined as follows:

$$P(z_{di} | z_{-di}, w, \alpha, \beta) \propto P(z_{di} | \hat{\theta}_d) P(w_{di} | z_{di}, \Phi)$$

(5)

where $z_{-di}$ are the latent topics except the one assigned for the $i$th word in $d$. In Eq. 5, only the $P(w_{di} | z_{di}, \Phi)$ component needs to access the original data. Hence, we integrate out $\hat{\theta}_d$ and the partially collapsed Gibbs update is as follows:

$$P(z_{di} | z_{-di}, w, \alpha, \beta) \propto \frac{(C_{DK}^{dK}, \alpha)}{\sum_{k}(C_{DK}^{dK} + \alpha)} P(w_{di} | z_{di}, \Phi)$$

(6)

where $C_{DK}^{dK}$ is the number of words assigned to topic $z_{di}$ in document $d$. Due to conjugacy of $\beta$ and $\Phi$, the update formula for $\phi_k$ is as follows:

$$P(\phi_k | w, z, \beta) \sim \text{Dirichlet}(C^{KW}_{k}, \beta)$$

(7)

where $C^{KW}_{k}$ is the number of word $w$ assigned to topic $k$ and $C^{KW}_{k}$ is an array with one element indicating the number of the corresponding word assigned to topic $k$. We write $P(w | z_{di}, \Phi)$ in the exponential family form:

$$P(w | z_{di}, \Phi) = \phi_{z_{di}, w} = \exp(\sum_{w'} n_{d_{w'}^i} \log \phi_{z_{di}, w'})$$

(8)

where $n_{d_{w'}^i} = \mathbb{I}[w' = w_{di}]$. Since the sampling algorithm interacts with the corpus only by the sufficient statistics for conditional probability $\exp(\sum_{w'} n_{d_{w'}^i} \log \phi_{z_{di}, w'})$, we privatize the sufficient statistics (i.e., $n_{d_{w'}^i}$) via the Laplace mechanism, resulting in privatized counts $\hat{n}_{d_{w'}^i}$.

### Table 1: Notations for FTM

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D$</td>
<td>the size of documents</td>
</tr>
<tr>
<td>$K$</td>
<td>the number of topics</td>
</tr>
<tr>
<td>$V$</td>
<td>the size of vocabulary</td>
</tr>
<tr>
<td>$\Phi$</td>
<td>the word distributions of topics</td>
</tr>
<tr>
<td>$\phi_k$</td>
<td>the word distribution of topic $k$</td>
</tr>
<tr>
<td>$\Theta$</td>
<td>topic distributions of documents</td>
</tr>
<tr>
<td>$\theta_d$</td>
<td>topic distributions of document $d$</td>
</tr>
<tr>
<td>$w$</td>
<td>the words vector of a document</td>
</tr>
<tr>
<td>$w_{di}$</td>
<td>$i$th word in document $d$</td>
</tr>
<tr>
<td>$z$</td>
<td>the topic assignment vector of a document</td>
</tr>
<tr>
<td>$z_{-di}$</td>
<td>the topic assignment of $i$th in document $d$</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>a topic index</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Dirichlet prior vector for $\Theta$</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Dirichlet prior vector for $\Phi$</td>
</tr>
<tr>
<td>$C^{DK}_{dK}$</td>
<td>the number of words assigned to topic $k$ in document $d$</td>
</tr>
<tr>
<td>$C^{KW}_{w}$</td>
<td>the number of word $w$ assigned to topic $k$</td>
</tr>
<tr>
<td>$M$</td>
<td>global topic model</td>
</tr>
<tr>
<td>$M^*$</td>
<td>updated global topic model</td>
</tr>
<tr>
<td>$M_p$</td>
<td>party $p$'s local topic model</td>
</tr>
</tbody>
</table>
We apply the “include/exclude” version of differential privacy, in which differing by a single entry refers to the inclusion or exclusion of that entry in the corpus. Since each counter $\hat{n}_{d_{iw}}$ is a sum of indicator vectors, it has $L_1$ sensitivity of 1. We have:

$$Y \sim \text{Laplace}(1/\varepsilon)$$

(10)

The above formula means randomly drawing a sample from the Laplace distribution with the location parameter 0 and scale parameter $1/\varepsilon$. Note that we only need to compute the privatized count $\hat{n}_{d_{iw}}$ once and it works as a proxy of the original count in the following sampling algorithms. Hence, no original data is exposed to the sampling algorithm. According to [47] and [13], it is easy to prove that such mechanism is $\varepsilon$-differentially private. After applying the Laplace mechanism, $\hat{n}_{d_{i}}$ is no longer sparse and the complexity of Gibbs sampling via Eq.6 increases from $O(\hat{n})$ to $O(K\hat{n})$ per word to $O(KW)$ per word, where $K$ is the number of topics and $V$ is the size of the vocabulary. It is easy to see that the private Gibbs sampling algorithm is unrealistically inefficient for real-life applications in which data sets are voluminous.

**ALGORITHM 1**: Private Metropolis Hastings

- **input**: local training data
- **output**: local topic model $M_p$

if it is the first global iteration then

for each document $d$ in local training data do

for each word $w_i$ in $d$ do

privatize $n_{d_{wi}}$: according to Eq.9 and threshold by $\tau$
end

end randomly assign a topic to each word in local corpus
else

build a word-topic alias table based on the new local model from the master
sample a topic for each word in local corpus according to the above word-topic alias table

for each local iteration do

build a doc-topic alias table according to Eq.11
build a word-topic alias table according to Eq.14

for each document $d$ in local corpus do

for each word $w_{di}$ in $d$ do

propose a topic $z_a$ with the doc-topic alias table;
update $z_{i}$ according to $z_a$ and Eq.13;
propose a topic $z_b$ with the word-topic alias table;
update $z_{i}$ according to $z_b$ and Eq.16;
end

end

sample $\Phi$ according to Eq.7

end

compose local model $M_p$ according to $c^{KW}_k$.

### 3.1.2 Private Metropolis Hastings

To improve the efficiency of MCMC sampling and make it applicable on massive dataset, we resort to private Metropolis-Hastings (MH), which is depicted in Algorithm 1.

Being the same as traditional MH, the private MH algorithm has two deliberately designed proposals for proposing a topic candidate for a word. The first proposal is the doc-topic proposal:

$$\Omega_{d}^{\tau} = \frac{(\hat{C}_{d_{k}^{DK}} + \alpha)}{\sum_{k'}(\hat{C}_{d_{k'}^{DK}} + \alpha)}$$

(11)

where $\Omega_{d}^{\tau}$ can be straightforwardly interpreted as the “strength” of the relation between $z$ and $d$.

For doc-topic proposal, the acceptance probability of topic transition from $z$ to $z'$ is:

$$\min\{1, \frac{P(z'|z_{-d}; w, \alpha, \beta)\Omega_{d}^{\tau}}{P(z|z_{-d}; w, \alpha, \beta)\Omega_{d}^{\tau}} \}$$

(12)

By replacing the component $P(z'|z_{-d}; w, \alpha, \beta)$ and $P(z|z_{-d}; w, \alpha, \beta)$ with Eq.6, the above acceptance probability is updated as follows:

$$\min\{1, \frac{(\hat{C}_{d_{z'}^{DK}} + \alpha)\hat{P}(w|z', \Phi)(\hat{C}_{d_{z}^{DK}} + \alpha)}{(\hat{C}_{d_{z}^{DK}} + \alpha)\hat{P}(w|z, \Phi)(\hat{C}_{d_{z'}^{DK}} + \alpha)} \}$$

(13)

where the hat notation means that the statistics of $w_{di}$ is removed from the corresponding value.

The second one is word-topic proposal, which is defined as:

$$\Omega_{w}^{\tau} = \frac{C_{z_{w}^{KW}} + \beta}{\sum_{z_{w}}(C_{z_{w}^{KW}} + \beta)}$$

(14)

where $\Omega_{w}^{\tau}$ can be straightforwardly interpreted as the “strength” of relation between topic $z$ and $w$.

For word-topic proposal, the acceptance probability of topic transition from $z$ to $z'$ is:

$$\min\{1, \frac{P(z'|z_{-d}; w, \alpha, \beta)\Omega_{w}^{\tau}}{P(z|z_{-d}; w, \alpha, \beta)\Omega_{w}^{\tau}} \}$$

(15)

By replacing the component $P(z'|z_{-d}; w, \alpha, \beta)$ and $P(z|z_{-d}; w, \alpha, \beta)$ with Eq.6, the above acceptance probability is updated as follows:

$$\min\{1, \frac{(\hat{C}_{z_{w}^{DK}} + \alpha)\hat{P}(w|z', \Phi)(\hat{C}_{z_{w}^{KW}} + \beta)}{(\hat{C}_{z_{w}^{DK}} + \alpha)\hat{P}(w|z, \Phi)(\hat{C}_{z_{w}^{KW}} + \beta)} \}$$

(16)

where the hat notation means that the statistics of $w_{di}$ is removed from the corresponding value.

The strategies of improving the sampling efficiency of private MH are twofold:

1. Improving the sampling efficiency of the proposals of Eq.11 and Eq.14. To achieve this goal, we build a doc-topic alias table and a word-topic alias table for the two proposals respectively according to the alias method in [54]. The key idea of the alias method is the construction of the alias table, which is illustrated by an example in Figure 2. During the construction process, the algorithm keeps moving “overfull” entries (entry one in the example) to the “underfull” entries (entry four in the example) in the table to make all the entries “exactly full”. At the meantime, it guarantees each entry has at most two kinds of entry index. With alias method, the
3.1.3 Topic-wise Normalization. The local model (i.e., the result of Algorithm 1) of a party \( p \) can be represented as a word-topic matrix \( M_p \), in which each cell stores the frequency count of the corresponding word and topic. The information in \( M_p \) should be transmitted to the master through network communication. As shown in Figure 3, transmitting \( M_p \) exposes word distribution of the training data on \( p \), since some important information may be recovered by \( M_p \) and deliberately designed language models. To solve this problem, we conduct topic-wise normalization and obtain the normalized word-topic matrix \( \hat{M}_p \), which is transmitted to the master. As \( M_p \) and \( \hat{M}_p \) result in exactly the same alias tables used in Algorithm 1, \( \hat{M}_p \) can be considered as the result of a lossless encryption mechanism, which protects the original word distribution of the training data on \( p \).

3.2 Master Computation

The duties of the master are twofold: integrating the local models from different parties and composing a new local model for each party. We first discuss how to integrate heterogeneous local topic models in Section 3.2.1. Then we discuss the approach of composing new local topic model for each party in Section 3.2.2. It is worth noting that master computation effectively handles topic models with different regularities and therefore is suitable for scenarios where data are not i.i.d. across parties.

3.2.1 Integrating Local Topic Models. Since the data of different parties are not necessarily i.i.d., the local models from different parties may contain different amounts of topics. The master is responsible for integrating these heterogeneous topic models. In order to compose a global model \( M^\ast \) based on the topics in local models, we rely on Weighted Jaccard Similarity to calculate the similarity between topics and merge the similar ones. The similarity between two topics \( z_i \) and \( z_j \) is defined as:

\[
\rho(z_i, z_j) = \frac{\sum_{l=1}^{m} \min(p_{z_i}^{z_j}, p_{z_j}^{z_i})}{\sum_{l=1}^{m} \max(p_{z_i}^{z_j}, p_{z_j}^{z_i}) + \sum_{l=m+1}^{T} p_{z_i}^{z_j} + \sum_{l=m+1}^{T} p_{z_j}^{z_i}}
\]

(17)

where \( p_{z_i}^{z_j} = (p_{w_1}^{z_j}, p_{w_2}^{z_j}, \ldots, p_{w_{m+1}}^{z_j}, \ldots, p_{w_L}^{z_j}) \) and \( P_{z_i} = (p_{w_1}^{z_i}, p_{w_2}^{z_i}, \ldots, p_{w_{m+1}}^{z_i}, \ldots, p_{w_L}^{z_i}) \) are vectors representing the top-L words distribution of topic \( z_i \) and topic \( z_j \). \( m (0 \leq m \leq L) \) indicates the count of common words in their top-L words. Two topics are considered as redundant if the similarity \( \rho(z_i, z_j) \) is beyond the threshold \( \xi \).

Based on above similarity metric, we detail the mechanism of integrating local models in Algorithm 2. The algorithm first concatenates two topic models (Line 2). Then it finds the redundant topic sets based on the Union-Find [14] algorithm (Line 2 ~ 11). For example, if \( (z_1, z_2) \) and \( (z_2, z_3) \) are considered as redundant based on Eq. 17, \( (z_1, z_2, z_3) \) will be taken as a disjoint topic set. For each topic set, we then merge the topics in the set to get the representative distribution (Line 12 ~ 16) by adding each topic distribution sequentially and do the normalization (In this case, the normalized distribution \( \frac{w_1 z_1 + w_2 z_2 + \cdots + w_L z_L}{w_1 + w_2 + \cdots + w_L} \) is chosen with \( z_1, z_2 \) and \( z_3 \) removed from \( M^\ast \)). Since the data of different parties are highly unbalanced, we assign different weight \( w_t \) to the topics based on
the data amount $n_i$ of different parties. Finally, we can obtain the global topic model $M^*$ (Line 18).

**ALGORITHM 2:** Integrating Local Topic Models  

**input:** global topic model $M$, local topic model $M_p$.  
**output:** updated global topic Model $M^*$.  
**begin**  
concatenate $M$ and $M_p$ into $M^B$;  
redundant topics $\mathcal{R} = \{\}$  
for each topic $z_i$ in $M^B$ do  
for each topic $z_j$ ($j > i$) in $M^B$ do  
calculate $\rho(z_i, z_j)$ with Eq. (17);  
if $\rho(z_i, z_j) \geq \xi$ then  
\hspace{1cm} Add ($z_i, z_j$) into $\mathcal{R}$  
end  
end  
end  
for each set $s$ in Union-Find($\mathcal{R}$) do  
for each topic $z_{si}$ ($i > 1$) in $s$ do  
\hspace{1cm} add $w_{si}z_{si}$ to $z_{si}$, remove $z_{si}$ from $M^B$;  
end  
normalize distribution $z_{si}$;  
end  
$M^* = M^B$  
**end**  
**return** $M^*$;  

3.2.2 Composing New Local Topic Models. Since the global topic model $M^*$ is large and comprehensive, some topics in $M^*$ is irrelevant to the data of certain parties. Hence, it is unnecessary to push all the information in $M^*$ to each individual party. In order to effectively reduce the communication cost, we compose a new local model that is compact enough to be pushed to the corresponding party. In order to take full advantages of the global model to facilitate local training, we employ meta-learning \cite{30} to transfer meta-level knowledge (i.e., the topics of $M^*$) as high-quality initialization for next-iteration local training. Specifically, we scan each topic $z_p$ in $M_p$, choose the most similar topic $z$ from the global topic model $M^*$, replace $z_p$ with $z$ in the new local topic model $M_p'$ and push it to $p$. The algorithm of composing new local models is presented in Algorithm 3.

**ALGORITHM 3:** Composing New Local Topic Models  

**input:** global topic model $M^*$, local topic model $M_p$  
**output:** new local topic model $M_p'$.  
**begin**  
for each topic $z_p$ in $M_p$ do  
\hspace{1cm} $z = \operatorname{arg max}_{z \in M^*} \rho(z, z_p)$;  
\hspace{1cm} if $\rho(z, z_p) \geq \xi$ then  
\hspace{1.5cm} replace $z_p$ with $z$ into $M_p'$;  
end  
remove $z$ from $M^*$;  
**end**  
**return** $M_p'$;  

We will show later that few global iterations are sufficient to obtain a good global topic model $M^*$. The low synchronization frequency improves FTM’s robustness to low bandwidth and network failures, which are more common in wide area networks than in data centers.

**ALGORITHM 4:** FTM Workflow  

**for each global iteration do**  
**for each client $p$ do**  
\hspace{1cm} train local topic model according to Algorithm 1  
\hspace{1cm} push local topic model $M_p$ to master  
end  
integrate local topic models and obtain the global topic model $M^*$ according to Algorithm 2  
**for each client $p$ do**  
\hspace{1cm} compose new local topic models for $p$ push new local topic model $M_p'$ to $p$  
end  
**end**  
**return** the global topic model $M^*$  

4 EXPERIMENTS  

In this section, we evaluate the performance of FTM in terms of both quantitative metrics and applications. In Section 4.1, we describe the experimental setup. In Section 4.2, we demonstrate the effectiveness of FTM in alleviating data scarcity. In Section 4.3, we demonstrate the utility of FTM in terms of different parameter settings. In Section 4.4, we gauge the communication cost of FTM. Finally, we show the necessity and the promising performance of FTM through a real-life application of ASR in Section 4.5.

4.1 Experimental Setup  

We assume that there are three parities denoted by $P_1$, $P_2$ and $P_3$, whose data are neither balanced nor i.i.d.. Specifically, $P_1$, $P_2$ and $P_3$ stores 29723, 59445 and 89169 documents respectively. A corpus containing another 29700 documents is used as the testing data. These baselines are trained through the LightLDA toolkit and the
number of topics has been tuned for each party to make them strong baselines.

4.2 Data Scarcity Alleviation

One major motivation of FTM is to alleviate the data scarcity problem faced by each individual party. Hence, one important question is whether the model trained by FTM is better than those trained by a single party relying on its own data. Figure 4 shows the comparison of FTM and different parties’ topic models in terms of the log likelihood of testing data. We observe that harnessing more data usually results in better topic models. By collectively utilizing data from all parties, FTM achieves the highest likelihood. For example, FTM-11 (i.e., FTM with $\varepsilon = 11$ and $\tau = 0.2$) demonstrates the log likelihood of $-2.74 \times 10^7$ while the best topic model trained by a single party is the one from $P_3$ and only achieves a log likelihood of $-3.03 \times 10^7$. Similar results can be observed with many other settings of $\varepsilon$ and $\tau$. We skip them due to space limitation. The result indicates that FTM is effective to alleviate data scarcity and generate high-quality topic models that cannot be obtained based upon a single party’s data.

4.3 Privacy Protection

The performance of FTM with different $\varepsilon$ reflects the utility of FTM after different levels of privacy protection. In Figure 5, we show the performance of FTM with regard to different $\varepsilon$ (i.e., the scale parameter for Laplace distribution) and $\tau$ (i.e., the threshold for sparsifying vector $\hat{\eta}_{di}$). As $\varepsilon$ increases, FTM usually achieves higher likelihood on the testing data. For example, FTM with $\varepsilon = 8$ and $\tau = 0.2$ achieves a log likelihood of $-4.59 \times 10^7$. When $\varepsilon$ increases to 11, FTM achieves a log likelihood of $-2.74 \times 10^7$. This observation is quite straightforward since $\varepsilon$ determines how much “noise” we add to the training data. In contrast, the effect of $\tau$ is more complicated, since it simultaneously affects the “noise” and the original data. When the “noise” is relatively moderate (e.g., $\varepsilon = 11$), a slightly higher $\tau$ (e.g., $\tau = 0.2$) will clap most of the noisy elements in $\hat{\eta}_{di}$ to zero and results in models with higher likelihood on testing data. Empirically, $\tau = 0.2$ demonstrates fairly good performance with moderate noise and we utilize it by default.

4.4 Communication Cost

Figure 6 presents the communication costs of conventional topic modeling and FTM with different $\varepsilon$. The baselines conventional topic modeling is a LightLDA model trained on a dataset consisting of the training data from $P_1$, $P_2$ and $P_3$. We observe that FTM converges quickly within several rounds of communication while conventional topic modeling demonstrates much slower speed of convergence. FTM with higher $\varepsilon$ demonstrates superior performance in terms of model quality and communication efficiency. When $\varepsilon$ is lower than 9, the final model of FTM is slightly worse than conventional topic modeling. Conventional topic modeling usually needs more than 300 rounds of communications to achieve the likelihood that can be achieved by FTM in less than 5 rounds. These results verify the superiority of FTM in low-bandwidth environment. Another interesting observation is that introducing moderate noise is beneficial for improve the quality of the model under training. When $\varepsilon$ is set to a value larger than 8, the models trained by FTM achieves higher likelihood than that trained by conventional topic modeling on original data.
components are evaluated by the standard metric Word Error Rate (WER) [27]. The lower the WER, the better the performance of the ASR system. A data set of 10-hour audio recordings is used for testing. The experimental results are shown in Table 2. We observe that all topic models are effective in reducing WER but the models trained on larger data are of higher quality. Even with the perturbation caused by privacy protection, FTM still achieves the best performance in terms of reducing WER, since harnessing comprehensive data significantly increases the quality of topic model. This application-oriented evaluation verifies our assumption that FTM can solve problems that plague real-life applications and improve their performance to a level that can not be achieved before.

<table>
<thead>
<tr>
<th>Models</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR without Topic Model</td>
<td>33.183%</td>
</tr>
<tr>
<td>ASR with Topic Model trained on ( P_1 )</td>
<td>31.401%</td>
</tr>
<tr>
<td>ASR with Topic Model trained on ( P_2 )</td>
<td>32.324%</td>
</tr>
<tr>
<td>ASR with Topic Model trained on ( P_3 )</td>
<td>33.035%</td>
</tr>
<tr>
<td>ASR with FTM-11</td>
<td>30.063%</td>
</tr>
</tbody>
</table>

5 CONCLUSION

In this paper, we propose a novel framework named Federated Topic Modeling (FTM) to solve two critical problems faced by industrial topic modeling: data scarcity and data privacy. By seamlessly combining techniques such as differential privacy, MCMC sampling and meta learning, FTM significantly alleviates the problem of data scarcity while providing a principled approach for protecting data privacy. With the federated architecture in FTM, a master and a series of parties work collectively to train high-quality topic models with low communication cost. Our quantitative experiments show that FTM has significant promise, as high-quality topic models can be trained in federated setting. Empirical evaluation of FTM on automatic speech recognition show that that it truly solves some real-life problems that have not been successfully handled before. Future work involves implementing more topic models based upon FTM.

ACKNOWLEDGMENTS

We are grateful to anonymous reviewers for their constructive comments. This work is partially supported by the National Science Foundation of China (NSFC) under Grant No. 61822201, U1811463 and 61532004. Yongxin Tong is the corresponding author of this paper.

REFERENCES


