An Efficient Approach for Cross-Silo Federated Learning to Rank

Yansheng Wang, Yongxin Tong, Dingyuan Shi, Ke Xu
SKLSDE Lab, School of Computer Science and Engineering, BDBC and IRI, Beihang University, China
{arthur_wang, yxtong, chnsdy, kexu}@buaa.edu.cn

***Abstract*—Traditional learning-to-rank (LTR) models are usually trained in a centralized approach based upon a large amount of data. However, with the increasing awareness of data privacy, it is harder to collect data from multiple owners as before, and the resultant data isolation problem makes the performance of learned LTR models severely compromised. Inspired by the recent progress in federated learning, we propose a novel framework named Cross-Silo Federated Learning-to-Rank (CS-F-LTR), where the efficiency issue becomes the major bottleneck. To deal with the challenge, we first devise a privacy-preserving cross-party term frequency querying scheme based on sketching algorithms and differential privacy. To further improve the overall efficiency, we propose a new structure named reverse top-K sketch (RTK-Sketch) which significantly accelerates the feature generation process while holding theoretical guarantees on accuracy loss. Extensive experiments conducted on public datasets verify the effectiveness and efficiency of the proposed approach.**

I. Introduction

In the last decade, Learning-to-Rank (LTR) has witnessed tremendous success in information retrieval (IR) systems [1]–[3], especially commercial search engines such as Google and Bing. Traditional LTR relies on massive data accumulated from interactions between the search engine and millions of web users. However, most companies except very few search engine giants do not have the privileges of generating sufficient training data by themselves. Even worse, with more data regulations and laws like GDPR coming into force, it becomes illegal for these companies to freely share or exchange data with each other, resulting in the well-known data isolation (*i.e.*, data fragmentation) problem [4]. Hence, how to break the barriers between data silos to train effective LTR models for enterprise search is still an open problem.

In this paper, we propose a framework named Cross-Silo Federated LTR (CS-F-LTR), which coordinates multiple companies (*i.e.*, silos or parties) to train a powerful LTR model without exchanging raw training data between them. In CS-F-LTR, both the documents and the queries necessary to train the model are distributed among different parties. Each party generates training instances in collaboration with the others while the documents and queries of each party are only locally stored for privacy protection. Unlike the generic federated learning setting where data are either partitioned horizontally or vertically [4], training data is *cross-partitioned* in our setting (see Fig. 1). As each training instance in LTR is correlated with a document and a query simultaneously, which may be owned by different parties, the feature generation may require collaboration between any two parties in cross-partitioned data settings.

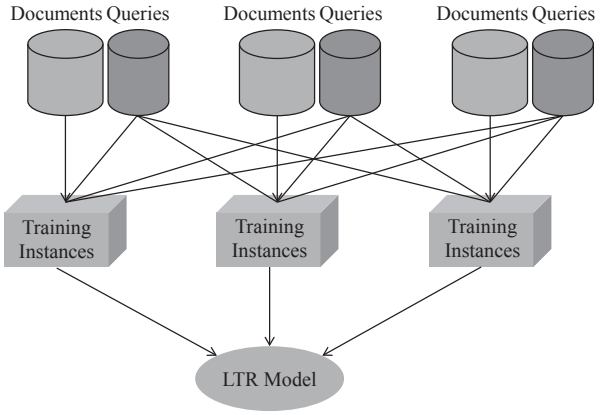


Fig. 1: An illustration of cross-partitioned data in federated LTR. Cross partition differs from horizontal and vertical partition in that each instance in the training data is generated by linking two components, queries and documents, which are also distributed across parties (silos).

Such unique data partition characteristics raise the major bottleneck in federated LTR: the efficiency issue. For one thing, encryption-based privacy-preserving schemes can be very low in efficiency and flexibility, as the cross-partition of data results in frequent interactions between parties. For LTR tasks, large scale of secure indexes needs to be built for document corpus, while the encrypted queries should be executed very frequently, which may take extremely high space and time cost. More practical ways are needed to compromise data privacy for efficiency while the privacy loss should still be controlled. For another, frequent interactions between cross-partitioned data can still bring high computation and communication cost even without encryption, which are also intrinsic bottlenecks in generic federated learning. Naïve solutions without special data structure designs will enumerate all the documents for a single party's single query in feature generation of federated LTR, which brings unacceptable querying time and communication overhead especially when the number of documents is large. Therefore, designing optimization techniques to improve the overall efficiency in federated LTR is vital.

2375-026X/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDE51399.2021.00102

In this paper, we focus on addressing the efficiency challenges above and our main contributions are summarized as follows:

- We formally define the federated LTR problem that has unique cross-partitioned data setting. We further identify its primary challenge as the efficiency problem and propose a solution framework named CS-F-LTR.
- To achieve symmetrical privacy with high efficiency in the feature generation of federated LTR, we propose a general term frequency querying scheme with sketching and differential privacy techniques, which has theoretical guarantees on both privacy and accuracy loss.
- To further optimize the overall efficiency of the algorithm, we propose a novel sketching structure named reverse top-K sketch (RTK-sketch), which can reduce both the querying times and communication cost while holding a theoretical guarantee of approximation.
- We evaluate the performance of our methods on public datasets and extensive experimental results validate the effectiveness and efficiency of our solution.

The rest of this paper is organized as follows. We review related work in Sec. II, formally define the problem in Sec. III, and elaborate on the term frequency querying scheme and the optimization approach in Sec. IV and Sec. V, respectively. The experimental results are presented in Sec. VI and we conclude this paper in Sec. VII.

II. Related Work

Our work is related to the following categories of research.

Privacy-Aware Information Retrieval. Information retrieval systems usually involve large scale data both from client side and server side that can support well-performed LTR models. The client-side browser query logs contain clickthrough of users that can reflect their daily online behaviors. Therefore privacy issues often occur, such as the famous privacy leak of AOL data [5]. Most existing privacy-aware LTR researches focus on protecting the client-side privacy from malicious servers. An early ideal model, private information retrieval (PIR) which aims to protect query privacy can be found in [6] but its requirement of duplicating the database makes it impractical in real scenarios. Some more practical approaches choose to obfuscate the queries by forging queries [7], generating cover queries [8] or injecting noise under differential privacy [9]. Symmetrically-private information retrieval (SPIR) was first defined in [10], where server-side data privacy is also considered. The most widely recognized methods are the secure keyword search schemes like searchable symmetric encryption (SSE) [11] and order preserving encryption (OPE) [12]. In [13], the authors also consider the data sharing scenario for multiple data owners and propose an OPE-based solution. However, these approaches only work for simple keyword searching rather than LTR. A few works consider privacy-aware LTR but is restricted to specific classifiers (like tree ensembles [14]). Different from previous work, in federated LTR, we consider protecting privacy among different parties during the collaborative learning process. A symmetrical privacy should be achieved for any two parties where the query privacy and document privacy are both important. In such a scenario, the prevailing encryption-based methods can be very low in efficiency.

Federated Learning. Federated learning (FL) [15], [16] was first proposed by Google for privacy-aware collaborative learning among android users and the definitions are generalized in [4]. It can be divided into cross-silo and cross-device federated learning [17]. The cross-silo setting naturally fits the business-to-business (B2B) scenarios where each silo can be a company or organization while the cross-device setting corresponds to the business-to-customer (B2C) mode. In both settings, privacy protection often becomes the core issue and encryption-based methods like secret sharing [18] and noisy-based methods like differential privacy [19]–[21] have been applied. The difference is that cross-device FL often involves huge number of users thus the communication cost can be a bottleneck while cross-silo FL only has a few parties (usually less than 10). Our paper is based on the cross-silo setting, as we assume each party to be an enterprise that hopes to build a global ranking model for certain specialized web search. In such a setting, the computation cost should be taken care of because each party as an enterprise has much more data than personal devices and encryption-based solutions can be very time-consuming. The unique feature of our problem is the cross partition of raw data, which has never been studied in FL before. A recent work [22] considering FL in online LTR framework is also relevant to our work. But it considers the cross-device FL and the data is horizontally partitioned, which makes existing solution frameworks of FL applicable.

Sketching Algorithms. To improve efficiency and to protect privacy, we propose an approach based on the sketch [23], which is a data structure for approximate statistical estimation in large scale streaming data. Count sketch [24] and Count-Min (CM) sketch [25] are two commonly used sketches for point queries like frequency estimation of a single term in a data stream. Many efforts have been made to optimize the efficiency and accuracy by proposing new data sketches [26]–[28]. The structure has also been applied to many areas, such as gradient compression in large scale machine learning [29] and heavy hitter discovery [30]. Sometimes it can also work with differential privacy (DP) [31], which aims to protect individual's privacy by injecting noise to aggregation results. In [32], the authors prove that Count sketch without additional noise can satisfy the notion of DP under strong assumptions. In this paper, we first modify the traditional sketching algorithms with new DP notions to realize privacy-preserving feature generation in federated LTR. Afterwards we devise a novel sketch to optimize the computation and communication efficiency.

III. Problem Statement

In this section, we formally define the problem setting of cross-silo federated LTR. Then we will define the reverse

top-K document query, which is fundamental to solving the problem.

A. Cross-Silo Federated LTR Setting

We first briefly explain the cross-silo setting in the context of learning to rank (LTR).

Suppose a federation $\mathcal{F}$ consists of N parties (enterprises), $\mathcal{F} = \{P_1, P_2, \cdots, P_N\}$. Each party P_i holds a collection of documents $\mathcal{D}_i = \{d_{i,1}, d_{i,2}, \cdots, d_{i,|\mathcal{D}_i|}\}$ as well as a collection of queries $\mathcal{Q}_i = \{q_{i,1}, q_{i,2}, \cdots, q_{i,|\mathcal{Q}_i|}\}$. Hence $\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}_i$ and $\mathcal{Q} = \bigcup_{i=1}^{n} \mathcal{Q}_i$. For each document d and query q, a relevant score $R(d, q)$ can be calculated, indicating the relevance between them. Note that each party only has access to relevance scores between its own documents and queries, though a query from party P_i can still be relevant to a document from party P_j.

As with other machine learning problems, to learn a ranking model, training data $(\boldsymbol{X}, \boldsymbol{Y})$ need to be generated, where $\boldsymbol{X}$ are features extracted from raw data D and Q, and $\boldsymbol{Y}$ are mapped from the relevance scores. In our work, we assume a feature extractor $\Phi : \mathcal{D} \times \mathcal{Q} \mapsto \mathbb{R}^s$ is given, which generates an s-dimensional feature vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_s)$ from a query q and its associated document d. Of our particular interest are features that involve both q and d, such as term frequency (TF), BM25 [33] and LMIR [34]. The labels are drawn from $0, 1$ and 2, representing "irrelevant", "relevant" and "highly relevant". We say that a sample is positive if the label is 1 or 2 otherwise negative.

Without federated learning, party P_i builds its own training data $\boldsymbol{X}_i, \boldsymbol{Y}_i$ from its own raw data $\mathcal{Q}_i$, $\mathcal{D}_i$. The training data can further be transformed to fit different ranking models, *e.g.*,, pair-wise or list-wise models. For simplification, we consider $(\boldsymbol{X}_i, \boldsymbol{Y}_i)$ as the final input of ranking model $\mathcal{M}_i(\boldsymbol{\theta}, \boldsymbol{x})$. Then the learning process can be a standard empirical risk minimization problem, *i.e.*,

$$\boldsymbol{\theta}_{opt} = \arg\min_{\boldsymbol{\theta}} E_{(\boldsymbol{x},\boldsymbol{y})\in(\boldsymbol{X}_i,\boldsymbol{Y}_i)}[L(\mathcal{M}_i(\boldsymbol{\theta}, \boldsymbol{x}), \boldsymbol{y})] \quad (1)$$

where L is the loss function. In the cross-silo federated learning setting, the locally generated data (especially positive instances) are insufficient thus each party needs to generate new feature vectors $\boldsymbol{X}'_i$ with other parties' collaboration.

For ease of coordination among parties, we assume a centralized server in the learning process. We also assume the server and each party are honest-but-curious (or semi-honest), *i.e.*, they will follow the protocol honestly and will not tamper with intermediate data, but will try to infer any sensitive information about each party from the available data. The objective of cross-silo federated learning is to efficiently train an effective global model from the data partitioned across parties while preserving data privacy during interactions among parties and the server.

Based on the generic setting above, we now formulate our problem of federated LTR as follows.

Definition 1 (Federated LTR Problem)**.** *Given a federation $\mathcal{F}$ with N parties, the purpose of federated LTR is to learn a global ranking model $\mathcal{M}$ collaboratively among all parties, where each party P_i generates an augmented dataset $\boldsymbol{X}'_i$ besides its own training set $(\boldsymbol{X}_i, \boldsymbol{Y}_i)$, meanwhile the following conditions should be satisfied:*

- ***Privacy:*** *During the interactions between any two parties or between the server and any party, the privacy leakage of each $\mathcal{D}_i$ and $\mathcal{Q}_i$ is controlled.*
- ***Effectiveness:*** *The collaboratively trained model $\mathcal{M}$ has better generalization than each individually trained model $\mathcal{M}_i$.*

Note that the privacy and effectiveness conditions are aligned with those in generic cross-silo federated learning. However, due to the cross partition of data in this problem, no existing FL methods can be directly applied. Next, we will show that a fundamental querying operation that we called the reverse top-K document query is the key to solving the federated LTR problem.

B. The Reverse Top-K Document Query

As we can see, the cross-partitioned data in federated LTR makes the feature generation much more challenging than standard learning tasks. Thus the key to effectively train LTR models is to effectively generate sufficient and high-quality training data with raw data from different parties. To generate useful and widely recognized features in LTR such as BM25 [33] and LMIR [34], the term frequency (TF) query is necessary. We formally define the cross-party TF as below.

Definition 2 (Cross-party TF)**.** *Suppose $P_i, P_j \in \mathcal{F}$, $d \in \mathcal{D}_j$, $q \in \mathcal{Q}_i$, and $t_1, t_2, \cdots, t_M$ are M terms in query q. Without loss of generality, we assume each document has L terms. The Cross-party Term Frequency of term t_k in document d is $TF_{i,j}(t_k, d) = \frac{TC_{i,j}(t_k,d)}{L}$, where $TC_{i,j}(t_k, d)$ the count of term t_k in document d.*

We assume the length of document is non-private, thus can be directly shared. So it is equal to calculating cross-party term counts TC. With the help of cross-party term frequency query, all the features can be generated for specific documents and queries. For example, the inverse document frequency (IDF) can be represented by

$$IDF_i(t_k) = \log \frac{\sum_{j=1}^{n} |\mathcal{D}_j|}{\sum_{j=1}^{n} \sum_{d\in\mathcal{D}_j} \mathbb{I}(TF_{i,j}(t_k, d) > 0)}$$

The BM25 score can also be written as

$$BM25_{i,j}(d, q) = \sum_{k=1}^{M} \frac{IDF_i(t_k) \cdot TF_{i,j}(t_k, d) \cdot (k_1 + 1)}{TF_{i,j}(t_k, d) + k_1}$$

where k_1 is a parameter.

In Sec. IV, we will discuss in detail how to realize the privacy-preserving cross-party TF.

However, calculating the cross-party TF is not enough to solve the problem. We can see that the generated feature matrix $\boldsymbol{X}'_i$ has no labels. If we generate samples for every possible document-query pairs between any two parties, there will be too much noisy data. In generic LTR the positive samples and

negative samples are highly skewed and positive samples are much more valuable data. Therefore we would like to find as many relevant documents for each query as possible and to exclude the irrelevant ones. Thus we define a new problem based on the TF query as below.

Definition 3 (Reverse Top-K Document Query)**.** *Suppose P_i has a single query term t. Suppose the document owner P_j has n documents $\mathcal{D} = (d_1, d_2, \cdots, d_n)$ and each document d_p has m terms $(t_{p,1}, t_{p,2}, \cdots, t_{p,m})$. Let $TC(t, d_p)$ denote the term count of t in document d_p. The reverse top-K document query problem is to find K documents in $\mathcal{D}$ which has the K largest $TC(t, d_p)$.*

Here we use term counts for simple calculation of relevance, and it can be replaced by any other TC-based metrics like BM25. A naive solution is to enumerate all the documents for the query term and rank their relevance scores to get the top-K relevant ones. But with a large number of parties and documents, such solution can be very low in efficiency. In Sec. V, we will concentrate on designing novel optimization techniques to improve both the computation and communication efficiency.

After the reverse top-K document query, each party will obtain the augmented data with positive labels. Combined with their local data, it will become a normal horizontal federated learning problem and we will apply a simple round-robin distributed SGD to train the LTR model while other sophisticated methods are also compatible. Note that the main challenges of federated LTR lie in the feature generation process. After we address the challenges, existing general FL methods can be simply applied which will not be our focus. Therefore, in the rest of our paper, we will concentrate on realizing privacy-preserving and efficient reverse top-K document query.

IV. Privacy-preserving Cross-Party Term Frequency Query

This section introduces our sketch and differential privacy-based scheme to realize privacy-preserving cross-party term frequency query, which is basic to reverse top-K document query. The objective is to calculate cross-party TF while the privacy of both parties (*i.e.*, document-side and query-side privacy) is protected. In such a multi-party scenario, where every two parties have to query each other's documents for many times, tradition encryption-based methods will be low in efficiency and flexibility. Therefore, we devise a sketch-based scheme, considering that the data structure has the following advantages. First, it is reusable after construction. Adding new parties will not take extra cost for other parties. Second, it is efficient both in memory and computation. The space cost can be linearly reduced meanwhile answering each query takes constant time. Third, it can hide information naturally, as hash functions are used to encode the data. With some further modifications, strong privacy guarantees can be met. Besides, only the sketches instead of the whole dataset of documents need to involve in the interactions in the learning process which can be safe and convenient for companies whose raw data cannot even be accessed by APIs of federated learning.

We will first introduce some preliminaries before elaborating on the details of our approach.

A. Preliminaries

We first introduce the sketch and the privacy requirements in the context of cross-party TF query.

The sketch is a certain class of streaming summaries, where a stream can be represented by a multiset d (like document) with terms t (like words) from $\mathcal{T}$ (*i.e.*, the dictionary). The sketch in our paper specifically refers to linear sketches that are data structures which can be represented as a linear transform of the input multiset. They are also defined for particular set of queries.

We use the classical Count Sketch [24] as a standard sketch in our paper. It can also be replaced by other sketches like Count-Min (CM) Sketch [25] or other state-of-the-arts. The Count Sketch is designed for point query of term frequency and can be represented by a $z \cdot w$ table. The encoding process requires two sets of hash functions, $\mathcal{H} = \{h_1, h_2, \cdots, h_z\}$ and $\mathcal{G} = \{g_1, g_2, \cdots, g_z\}$ randomly sampled from pairwise independent hash function families with $h_i : \mathcal{T} \to [1, w] (w \ll |\mathcal{T}|)$ and $g_i : \mathcal{T} \to \{+1, -1\}$. The encoding of each term $t \in d$ follows $\forall 1 \leq a \leq z$,,

$$C_d(a, h_a(t)) \leftarrow C_d(a, h_a(t)) + g_a(t) \tag{2}$$

After the encoding of document d, we will get a table $C_d(\cdot, \cdot)$ with size $z \cdot w$. The point query of term frequency t follows:

$$\hat{f}_t = f_C(d, t) = \text{median}\, C_d(a, h_a(t)) \tag{3}$$

As the querying process in sketch requires multiple hash functions, we can simply obfuscate some of them to preserve the query-side privacy. To further preserve the document-side privacy from adversarial queriers we can define the ϵ-Differential Privacy (ϵ-DP) for point queries, which depends on sketch and can be a bit different from the general ϵ-DP definition.

Definition 4 (ϵ-Differential Privacy (ϵ-DP))**.** *A random algorithm $\mathcal{A}$ satisfies ϵ-DP, if $\forall$ neighboring documents d, d' that differ from one term, $\forall$ point queries with term t and possible outputs o of $\mathcal{A}$,*

$$Pr[\mathcal{A}(f_C(d', t)) = o] \leq e^{\epsilon} Pr[\mathcal{A}(f_C(d, t)) = o]$$

Our goal is to make the point query results satisfy ϵ-DP so that no privacy information from the documents will be inferred.

B. Method

Next we present our cross-party TF query scheme in detail. Suppose party P_i has a term t (which can be a term from one of its queries) and it wants to find the term frequency of t in document d owned by party P_j. The querying process operates in three steps.

- **Step 1: Sketch Construction.** First, party P_j constructs the sketch for document d. The sketch construction is conducted before any interaction among parties. Each party uses the same hash functions for sketch construction so that it can support the queries from any other parties. The hash functions can be keyed where the private keys are securely generated (*e.g.*, with Diffie-Hellman key agreement) so that they can be hidden from the server. We build the Count Sketch for each document d following (2), where the terms are words from the vocabulary set $|\mathcal{T}|$ and we will get a table of $z \cdot w$ slots. The document d has L terms in total and it takes $O(z \cdot L)$ times of hashing to construct the sketch. The sketches of each document from each party are stored privately and can only be accessed by queries on the frequency of specific terms.
- **Step 2: Hashing With Obfuscation.** After sketch construction, party P_i has to hash its term t with the z hash functions from $\mathcal{H}$ in order to query its TC. To protect the privacy of the terms, we will not calculate $h_a(t)$ for every $1 \leq a \leq z$ as the normal Count sketch does. Instead, we randomly pick z_1 hash functions from $\mathcal{H}$ and calculate their hashes on t. For the other $z - z_1$ functions, the input is randomly sampled from $\mathcal{T}$. Formally, after this hashing and obfuscating process, party P_i will get a z-dimensional vector $(h_a^{(i)}(t))_{1 \leq a \leq z}$ encoded by a private index set with length z_1, *i.e.*,

$$h_a^{(i)}(t) = \begin{cases} h_a(t), a \in PV_i \\ \\ h_a(t'), a \notin PV_i, t' \sim \mathcal{T} \end{cases} \quad \forall 1 \leq a \leq z \tag{4}$$

  where PV_i is the set containing the first z_1 values from a random permutation of $\{1, 2, \cdots, z\}$, which stands for the hash indexes of the real querying terms. A smaller fraction of $\frac{z_1}{z}$ will result in stronger protection of the querying terms. There is also a trade-off between privacy and accuracy loss as we can set smaller z_1 to achieve higher privacy level while the confidence of the querying results will be reduced. After finishing the hashing, P_i will send the obfuscated hash vector to the server, and the server will send it to P_j for further processing.
- **Step 3: Result Perturbation.** In this step, P_j will receive the z-dimensional vector from the server. Afterwards, it will conduct the query on Count sketch with each $h_a^{(i)}(t)$ and get $C_{d.\mathcal{H}}(a, h_a^{(i)}(t))$ for $1 \leq a \leq z$. A direct release of such results can also be risky, as an adversarial querier will send some sensitive terms to infer their distribution in other party's documents. Therefore, we hope to design ϵ-DP mechanism to perturb the results so that the privacy of documents can be preserved. Following the Laplace Mechanism [31], our perturbing method is rather simple as below,

$$\tilde{C}_d(a, h_a^{(i)}(t)) = C_d(a, h_a^{(i)}(t)) + \tilde{N} \tag{5}$$

  where $\tilde{N} \sim Lap(\frac{1}{\epsilon})$ and ϵ is the privacy budget. We only sample one $\tilde{N}$ for all z hashing results and we will later prove that the mechanism satisfies ϵ-DP. Then P_j will send the perturbed sketch results to P_i through the server. After receiving the corresponding results of the hashed values, party P_i only needs to recover the correct ones with its private key PV_i. The final query results of TC of t will be

$$\hat{f}_t = \underset{a \in PV_i}{\text{median}} \tilde{C}_d(a, h_a(t)) \tag{6}$$

  Although the frequencies of items are not precise due to the noise brought by the sketch, without the DP noise, a malicious querier can still infer the approximated frequency distribution of words of a document which can result in privacy leaks. We parameterize the privacy protection level by injecting a Laplacien noise with variance $\frac{1}{\epsilon}$ so that any intention of inferring a word's frequency of a document can be obfuscated by ϵ-DP.

Algorithm 1: Cross-party TF: Querier

input : t: the term for frequency query
$\mathcal{H} : \{h_a(\cdot) | 1 \leq a \leq z\}$, the agreed hash functions
output: $\hat{f}_t$: the estimated frequency of t

```
Q ← Empty vector;
PV ← Randomly generated z_1 hash indexes;
for 1 ≤ a ≤ z do
    Generate hash according to 4;
    Q.append(hash);
Send Q to server;
Receive querying results F̃_Q from server;
F̃_{Q,real} ← {f_a ∈ F̃_Q | a ∈ PV};
f̂_t ← Estimator(F̃_{Q,real});
return f̂_t
```

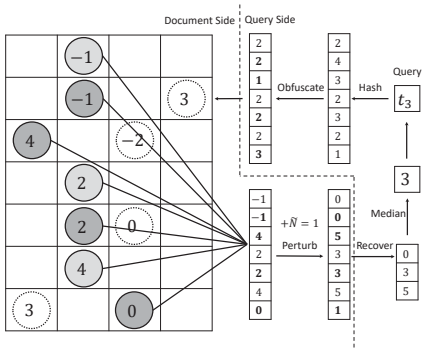


Fig. 2: The TC query of term t_3 from the sketch, where both query-side and document-side privacy are preserved.

Algorithm 1 and Algorithm 2 illustrate the operations at the querier (*i.e.*, P_i) and the document owner (*i.e.*, P_j), respectively. Next we will make analysis on the privacy and accuracy loss.

Algorithm 2: Cross-party TF: Document Owner

input : d: the document containing l_d terms
Q: hash values of querying term t
$\mathcal{H} : \{h_a(\cdot)|1 \le a \le z\}$, the agreed hash functions
ϵ: privacy budget
output: None

$C \leftarrow Constructor(d, \mathcal{H})$;
Receive querying vector Q from server;
$\tilde{F}_Q \leftarrow$ Empty vector;
Sample $\tilde{N} \sim Lap(\frac{1}{\epsilon})$;
for $1 \le a \le z$ **do**
  $f_a \leftarrow C.find(Q[a]) + \tilde{N}$;
  $\tilde{F}_Q.append(f_a)$;
Send $\tilde{F}_Q$ to server;
return *None*

C. Theoretical Analysis on Privacy and Accuracy Loss

We make analysis on the privacy and accuracy loss of our cross-party term frequency querying scheme.

1) Guarantees on Privacy:

Theorem 1. *The estimator in Eq.(5) satisfies ϵ-DP for any point queries.*

Proof. Suppose the length of document d' is $L' = L + 1$. Document d and d' coincide in the first L terms and d' has an additional term t'. According to the pairwise independence of hashing families, we have

$$Pr[h_a(t) = h_a(t')] \le \frac{1}{range(h_a)} = \frac{1}{w}, \forall t \ne t'$$

Then, with fixed i and PV_i, we have $\forall$ querying terms $t \ne t'$ and $\forall a \in PV_i$,

$$C_{d'}(a, h_a(t)) = C_d(a, h_a(t)) + RX_a \cdot g_a(t)$$

where each RX_a is *i.i.d* drawn from a Bernoulli distribution with $Pr[RX_a = 1] \le \frac{1}{w}$. So we have

$f_C(d', t) = \underset{a \in PV_i}{\text{median}} C_{d'}(a, h_a(t)) = \underset{a \in PV_i}{\text{median}}(C_d(a, h_a(t)) + RX_a \cdot g_a(t)) = \underset{a \in PV_i}{\text{median}}(C_d(a, h_a(t)) + RY = f_C(d, t) + RY$, where $RY \in \{+1, 0, -1\}$.

With $a_0 = \arg\text{median}_{a \in PV_i}(C_d(a, h_a(t))$, we have $Pr[RY = 1 \vee RY = -1] = Pr[f_C(d', t) = h_{a_0}(t) + 1 \vee f_C(d', t) = h_{a_0}(t) - 1] \le Pr[RX_{a_0} = 1] \le \frac{1}{w}$. After injecting the Laplacian noise $\tilde{N} \sim Lap(\frac{1}{\epsilon})$, *i.e.*, $\mathcal{A}(f_C(d, t)) = f_C(d, t) + \tilde{N}$, we have

$$\frac{Pr[\mathcal{A}(f_C(d', t)) = o]}{Pr[\mathcal{A}(f_C(d, t)) = o]} = \frac{Pr[\tilde{N} = o - f_C(d, t) - RY]}{Pr[\tilde{N} = o - f_C(d, t)]}$$
$$= Pr[RY = 0] + Pr[RY = 1] \cdot \frac{Pr[\tilde{N} = o - f_C(d, t) - 1]}{Pr[\tilde{N} = o - f_C(d, t)]}$$
$$+ Pr[RY = -1] \cdot \frac{Pr[\tilde{N} = o - f_C(d, t) + 1]}{Pr[\tilde{N} = o - f_C(d, t)]}$$
$$\le 1 - \frac{1}{w} + \frac{1}{w} \cdot e^{\epsilon} \le e^{\epsilon}$$

Otherwise, for $t = t'$, we have $\forall a \in PV_i$, $C_{d'}(a, h_a(t)) = C_d(a, h_a(t)) + 1$. Therefore, $f_C(d', t) = f_C(d, t) + 1$, and we can simply have $\frac{Pr[\mathcal{A}(f_C(d', t)) = o]}{Pr[\mathcal{A}(f_C(d, t)) = o]} \le e^{\epsilon}$, and the theorem follows. □

2) Guarantees on Accuracy Loss: The frequency estimator of Count sketch [24] gives unbiased results with a variance of $\frac{F_2}{w}$ where $F_2 = \sum_{1 \le k \le l_d} f_{t_k}^2$. To further decrease the utility loss, we can consider the skewness of data. The frequency of words in the documents often follow the Zipf's law [1], and by following [35], we can reduce F_2 to $F_2^{Res} = \sum_{r \le k \le l_d} f_{t_k}^2 \le \frac{c_z^2 (r-1)^{1-2\zeta}}{2\zeta - 1}$ where $f_i = \frac{c_z}{i^\zeta}$ is the frequency of the i^{th} most frequent item under Zipf's law. Then the error bound of the TC estimation of the single term is as below.

Theorem 2. *For a single term t, if z_1 is set to $O(log(\frac{1}{\delta}))$, then with probability at least $1 - \delta$, we have the TC estimation of the term has a error bounded by*

$$|\hat{f}_t - f_t| \le \sqrt{\frac{16}{\epsilon^2} + \frac{64}{w} \cdot F_2^{Res}}$$

Proof. According to (5), We have

$$C_a(t) = f_t + \sum_{t' : g_a(t') = g_a(t)} g_a(t) g_a(t') f_{t'} + Lap(\frac{1}{\epsilon})$$

According to the expectation and variance of count sketch estimator and laplacian variable, we have $E[C_a(t)] = f_t$ and $Var[C_a(t)] = \sum_{t' : g_a(t') = g_a(t)} f_{t'}^2 + \frac{2}{\epsilon^2}$. By assuming the Zipf's distribution of term frequency, with constant probability $\frac{7}{8}$ over the choice of hash functions, none of the $r = \frac{w}{8}$ heaviest items collide with t in any given row. Thus $E[\sum_{k>r} f_{t_k}^2] = \frac{F_2^{Res}}{w}$. By Markov inequality, $Pr[\sum_{k>r} f_{t_k}^2 \le \frac{8F_2^{Res}}{w}] \ge \frac{7}{8}$. Thus we have

$$Pr[Var[C_a(t) \le \frac{8F_2^{Res}}{w} + \frac{2}{\epsilon^2}]] \ge \frac{7}{8} \quad (7)$$

By Chebyshev inequality,

$$Pr[|C_a(t) - f_t| \ge \sqrt{\frac{64F_2^{Res}}{w} + \frac{16}{\epsilon^2}}] \le \frac{1}{8} \cdot \frac{Var[C_a(t)]}{8F_2^{Res}/w + 2/\epsilon^2} \quad (8)$$

[1] https://en.wikipedia.org/wiki/Zipf%27s_law

By combining (7) and (8), we have

$$Pr[|C_a(t) - f_t| \leq \sqrt{\frac{64F_2^{Res}}{w} + \frac{16}{\epsilon^2}}] \geq 1 - \frac{1}{8} - \frac{1}{8} - \frac{1}{8} = \frac{5}{8}$$

Since the hashes are independent, by Chernoff bounds, we finally have

$$Pr[|\hat{f}_t - f_t| \geq \sqrt{\frac{64F_2^{Res}}{w} + \frac{16}{\epsilon^2}}] \leq e^{-O(z_1)}$$

□

For a query q with l terms, we use the following TC estimator $\hat{f}_q = \underset{a \in PV_i}{\text{median}} \sum_{1 \leq k \leq l} \tilde{C}_d(a, h_a(t_k))$. The error bound is below.

Theorem 3. *For a query q with length l, if we set $z_1 = O(log(\frac{1}{\delta}))$, then with probability at least $1 - \delta$, we have the TC estimation of the query has a error bounded by*

$$|\hat{f}_q - f_q| \leq \sqrt{\frac{16l}{\epsilon^2} + \frac{64l}{w} \cdot F_2^{Res}}$$

Proof. We first prove that for any two terms t_1 and t_2 and any $a \in PV$, $C_a(t_1)$ and $C_a(t_2)$ are independent. We have $C_a(t_1) = f_{t_1} + \sum_{t':g_a(t')=g_a(t_1)} g_a(t_1)g_a(t')f'_{t_1} + Lap(\frac{1}{\epsilon})$ and $C_a(t_2) = f_{t_2} + \sum_{t':g_a(t')=g_a(t_2)} g_a(t_2)g_a(t')f'_{t_2} + Lap(\frac{1}{\epsilon})$. Then we have

$$\begin{aligned}
&E[(C_a(t_1) - f_{t_1})(C_a(t_2) - f_{t_2})] \\
=&E[\sum_{g_a(t')=g_a(t_1)} g_a(t_1)g_a(t')f_{t'} \cdot \sum_{g_a(t'')=g_a(t_2)} g_a(t_2)g_a(t'')f_{t'}] \\
=&E[\sum_{i',i''} g_a(t_1)g_a(t')g_a(t_2)g_a(t'')ft'^2] = 0
\end{aligned}$$

Therefore, we have $E[\sum_{1 \leq k \leq l} C_a(t_k)] = \sum_{1 \leq k \leq l} E[C_a(t_k)] = \sum_{1 \leq k \leq l} f_{t_k}$ and $Var[\sum_{1 \leq k \leq l} C_a(t_k)] = \sum_{t',k:g_a(t')=g_a(t_k)} f_{t'}^2 + \frac{2p}{\epsilon^2}$. Following the proof of Theorem 2, we have

$$Pr[|\hat{f}_q - f_q| \geq \sqrt{\frac{16l}{\epsilon^2} + \frac{64l}{w} \cdot F_2^{Res}}] \leq e^{-O(z_1)}$$

□

V. Efficient Reverse Top-K Document Query

Based on the privacy-preserving term frequency query in Sec. IV, we will further study the reverse top-K document query in this section, which is essential to federated LTR. First, we will propose a NAIVE solution based on the TF query. We will find that it is low in efficiency and then will devise a new data structure named reverse top-K sketch (RTK-Sketch) to improve the efficiency. As computation and communication efficiency have always been bottlenecks for generic federated learning, our proposed solution will be meaningful to real applications.

A. NAIVE solution

The NAIVE solution is shown in Algorithm 3. It only works for the querier side as the document owner side does not need to do extra computation. Obviously, it is not efficient enough. To find the top-K relevant documents for a querying term, it has to enumerate all the documents in a party, which leads to n times of sketch queries. The time complexity is $O(zn)$ for a single querying term. Meanwhile, the server has to transmit the perturbed sketching results for all the n documents, which brings high communication overhead. With the increase of documents, the total computation and communication cost will become unacceptable. Next, we will introduce our optimization techniques.

Algorithm 3: NAIVE

input : Query term t, parameter K
output: K tuples of document id and term count

$Res \leftarrow \emptyset$;
for *document* $d_1, d_2, \cdots, d_n \in$ *document owner* **do**
  $c_i \leftarrow Query(d_i, t)$ according to Algorithm 1;
  $Res \leftarrow Res \cup \{i : c_i\}$;
$Res \leftarrow TopK(Res)$;
return *Res*

Algorithm 4: RTK-Sketch: Update

input : A document d with index id, parameter α
output: None

$T \leftarrow$ Build a standard sketch on d according to Algorithm 2;
for $i = 1, 2, \cdots, z$ **do**
  for $j = 1, 2, \cdots, w$ **do**
    $S[i][j].Insert(\{id : T[i][j]\})$;
    if $|S[i][j]| > \alpha K$ **then**
      $S[i][j].DeleteMin()$;

B. RTK-Sketch

In this part, we focus on using optimization techniques to reduce the query times and transmission data from $O(n)$ to $O(K)$ for a single term. The idea is to carry out more computation locally before the querying starts. We design a novel sketch-based data structure named reverse top-K sketch (RTK-Sketch). It maintains the top-$O(K)$ counts and document indexes in each cell of a standard sketch and uses intersection of the hashed cells to query a specific term. Besides higher efficiency, the structure is also flexible to use. If some party wants to update new documents or delete old documents, they only have to do incremental updates instead of re-constructing the whole sketch. Although the sketch returns the approximated top-K results, we will prove theoretically that it can still cover a constant ratio of true top-K documents. The details of the algorithm are as below.

Initialize. The initialization happens in the document owner side and the RTK-Sketch will replace the former n sketches for all the documents. It is also initialized by an array with z rows and w columns, indicating that we need z pairwise

Algorithm 5: RTK-Sketch: Query

input : Sketch S, query term t, parameter β
output: k tuples of document id and term count

$Cand \leftarrow \emptyset$;
$D \leftarrow dict()$;
for $i = 1, 2, \cdots, z$ **do**
  $H_i \leftarrow List(S[i][h_i(t)])$;
  for $(id, count)$ *in* H_i **do**
    $D[id].Append(count)$;
for id *in* D **do**
  if $|D[id]| \geq \beta z$ **then**
    $count \leftarrow Query(D[id])$ according to Algorithm 1;
    $Cand \leftarrow Cand \cup \{id : count\}$;
$Res \leftarrow TopK(Cand)$;
return Res

independent hash functions with range w for them. Different from Count Sketch where each grid in the sketch table is an integer representing the count after hashing, in RTK-Sketch a table cell is a list of document indexes and their counts. To realize faster deletion of minimal elements, we initialize each cell with a *Min-Heap*.

Update. The update algorithm is also for the document owner side, where each document is considered as streaming input to update the sketch. The details are shown in Algorithm 4. To update a new document d with index id in the RTK-Sketch S, we first apply the normal sketching algorithms such as Count Sketch following Algorithm 2 for each terms in d. We will get a table T with exactly the same number of rows and columns as S, while each element in T is an integer. Then we will insert the pair of $\{id : T[\cdot][\cdot]\}$ into S meanwhile we ensure that each Min-Heap has at most αK elements. The insertion and deletion take at most $O(\log \alpha K)$ time, thus the time complexity of updation becomes $O(wz \log \alpha K)$.

Delete. To delete a document with index id in the sketch, the document owner has to enumerate every grid of the sketch S and to remove the document in each Min-Heap. It takes $O(\alpha K)$ to find an element and $O(\log \alpha K)$ to delete it in a heap with size αK thus the time complexity of deletion becomes $O(wz\alpha K)$.

Query. The querier has to execute the reverse top-K document query for term t on the RTK-Sketch. The algorithm is shown in Algorithm 5. In the querying algorithm, we first enumerate every Min-Heap in the cells that t can be hashed to then every element in those Min-Heaps to build a dictionary that maps a document index to a list of its corresponding counts with different hash functions. We will filter out the documents that appear less than βz times. The rests are the top-αK documents for at least βz hash functions. We will put them into a candidate set and their corresponding counts come from the sketch querying operator following Algorithm 1 (*e.g.*, median

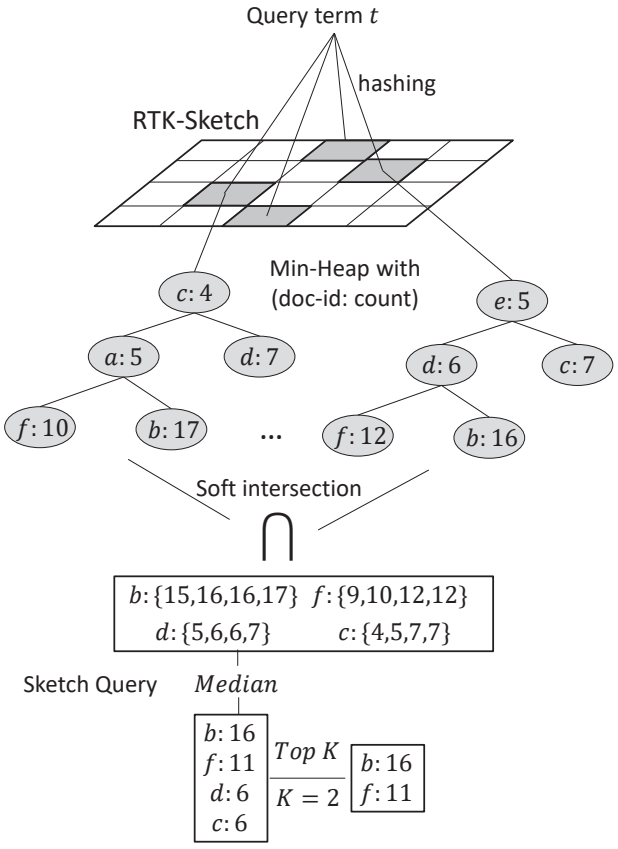


Fig. 3: An example of querying in RTK-Sketch. Each cell in the sketch table is a Min-Heap with document id and its corresponding count as elements. The soft intersection refers to that a document appears at least β fraction times among all the Min-Heaps.

for Count Sketch). Finally the querying results are the top-K documents of the candidate set. The time complexity of one query is $O(z\alpha K)$. An example is illustrated in Fig. 3.

C. Theoretical Analysis on Accuracy Loss

In this part, we will make theoretical analysis on the accuracy loss of top-K results. The accuracy loss on counts remains the same as in Sec. IV and we will abuse z_1 by z for simplification.

Theorem 4. *For a fixed query term t, suppose K' is the number of documents that appear both in the real top-k querying results and the results that Algorithm 5 returns. The expectation of cover rate $E_{CR} = \frac{E[K']}{K}$ will hold a constant lower bound if $z \geq (1 - \frac{1}{\eta^2})^{-O(n)}$ and $\beta < \sqrt{\frac{1}{z}}$ with $\eta = \frac{(\alpha-1)\sqrt{Lw}}{2\alpha kq}$.*

Proof. We fix the query term t and rank the term counts of t for all the n documents, where each document has L terms. Suppose the ranking result is $c_1 \geq c_2 \geq \cdots \geq c_K \geq \cdots \geq$

c_n with document indexes from 1 to n. Then the real top-k document querying result is $\{1, 2, \cdots, K\}$. If we take all the counts with document index i from sketch S, we will get z integers (though some of them might be removed by the heap) denoted by $c_{i,1}, c_{i,2}, \cdots, c_{i,z}$. Each count can be seen as the sum of the real count c_i and a random variable Δ_i which is the noise brought by sketching. And we have $E(\Delta_i) = 0$ and $Var(\Delta_i) = \sigma^2 = O(\frac{1}{w} F_2^{Res}) \approx O(\frac{L}{w})$ (assuming the residue of $O(L)$ terms are no larger than 1). By assuming the counts c_i satisfies Zipf's law with $c_1 = \frac{L}{q}$ where $q > 1$ is a constant parameter, we get $c_i = \frac{L}{iq}$. For the l^{th} hash function, by Chebyshev's inequality we have

$$\begin{aligned} &Pr[c_{\alpha K,l} \leq c_{K,l}] \\ &\geq Pr[\Delta_K \geq -\eta\sigma] \cdot Pr[\Delta_{\alpha K} \leq c_K - c_{\alpha K} - \eta\sigma] \\ &\geq (1 - \frac{1}{1+\eta^2}) \cdot (1 - \frac{\sigma^2}{(\frac{(\alpha-1)L}{\alpha K q} - \eta\sigma)^2}) \geq (1 - \frac{1}{\eta^2})^2 \end{aligned}$$

where $\eta = \frac{(\alpha-1)\sqrt{Lw}}{2\alpha K q}$.

Thus, the probability that $i^{th} (i \leq K)$ document is in the top-αK ranking list after sketching is $p_i = \Pi_{j=\alpha K+1}^{n} Pr[c_{\alpha K,l} \leq c_{j,l}] \geq Pr[c_{\alpha K,l} \leq c_{K,l}]^{n-\alpha K}$. Suppose event E_i represents that $i^{th} (i \leq K)$ document appears no larger than βz times among z sets $H_1, H_2, \cdots, H_z$. And with the tail bound of cumulative binomial distribution, we have

$$Pr[E_i] = \sum_{j=1}^{\beta z} \binom{z}{j} p_i^j (1-p_i)^{z-j} \leq e^{-2z(p_i-\beta)^2} (\beta < p_i)$$

If $p_i = \Omega(\sqrt{\frac{1}{z}})$ holds, $Pr[E_i]$ can be bounded by a constant. Therefore by ensuring that $z \geq (1 - \frac{1}{\eta^2})^{-4(n-\alpha K)}$ and $\beta < \sqrt{\frac{1}{z}}$, the expectation of cover rate $E_{CR} = \frac{\sum_{i=1}^{k} Pr[E_i]}{K}$ will have a constant lower bound. □

Remarks. When n is very large, the condition of the theorem will make z unreasonably large. Nevertheless, in our experiments with a large n and relatively small z the cover ratio still remains constant. The reason is that the data may be much more skewed than we assume. There may be many zeros in the residual terms of c_i and the probability that the counts of our top-K documents are larger than the residual terms after sketching can approach very closely to 1 (our lower bound will appear to be too loose for the residues). Then the exponential term $O(n)$ can be reduced to much smaller values, or even constants. In that case, the condition of z and β can be largely loosed.

VI. Experimental Evaluations

In this part, we will report our experimental results to verify the effectiveness and efficiency of proposed methods.

A. Dataset and Settings

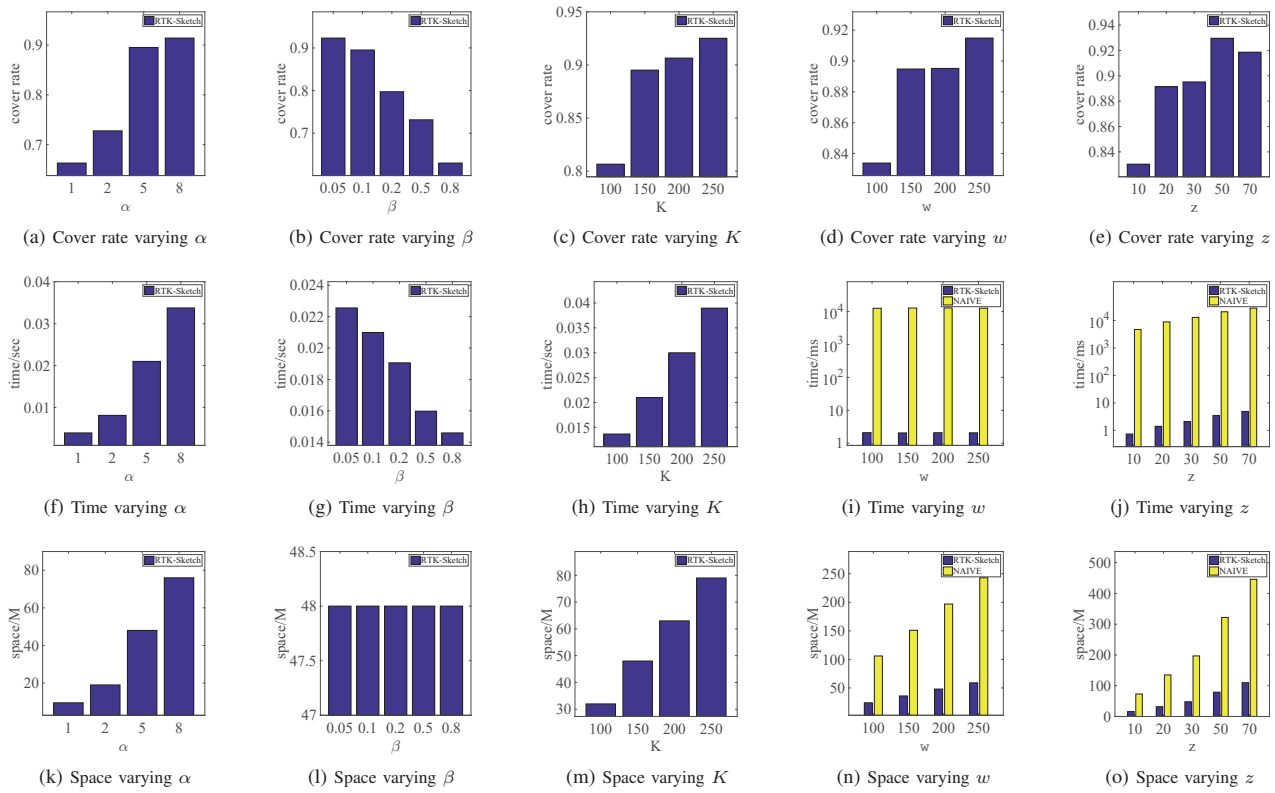
Most existing benchmark datasets like LETOR 4.0 [36] only contain extracted features rather than raw documents and queries, thus can not be used in our experiments. So we choose the MS MARCO Ranking dataset [2] and sample some subsets for our experiments. We assume the number of parties $N = 4$, and each party only has limited labeled querying results. Note that 4 parties are sufficient in cross-silo FL settings as each party can be an enterprise. We sample 4 subsets from MS MACRO, each contains 200 queries and 36,400 documents. Each document has about 1000 terms. We use the top100 ranking as the ground-truth in our dataset. The top10 documents are labeled by "highly relevant" (relevance score = 2) while top11-100 are "relevant" (relevance score = 1). The others are considered as "irrelevant" (relevance score = 0). For each party, we generate about 28,000 training instances. We also simulate an external test set with 32,000 instances by extracting 8,000 other instances from each party. Note that in real scenarios the evaluation process does not require data or model sharing among parties as each party holds a local model. The features we use include length, TF, IDF, TF-IDF, BM25, LMIR.ABS, LMIR.DIR and LMIR.JM of each document's body and title, which form a 16-dimensional vector for each instance. We generate 29,000 cross-party instances for each party. We use a simple linear classification for training a point-wise ranking model. As we focus on proposing a general LTR framework, we do not use more complicated features like PageRank, which can be considered as the intrinsic features of documents and do not require further privacy protection techniques. More complicated models like pair-wise ranking with tree models are also compatible as long as they need to conduct frequency queries on different documents. But we may not support the deep Seq2Seq models which directly generate feature vectors from raw data without frequency queries. Evaluation metrics used in our experiments include ERR, nDCG and nDCG@10.

We compare our approach, CS-F-LTR, with the following methods:

- Local: each party trains a local model only with its own dataset.
- Local+: each party trains a local model with both local and augmented data that are generated by cross-party queries between itself and others.
- Global: each party collaboratively train a global model only with their local data, like the horizontally federated learning does. However, we do not use any privacy protection techniques so the results are lossless.

To evaluate our optimization technique, *i.e.*, the RTK-Sketch, we compare its time and space costs with the NAIVE solution. We also show its overall performance varying the parameters, where the default parameter setting is $\alpha = 5, \beta = 0.1, w = 200, z = 30, K = 150, \epsilon = 0.5$. The hash function we used in the sketch is the MD5 algorithm. We implement the learning algorithms with MindSpore [3].

[2]https://www.msmarco.org/dataset.aspx
[3]https://www.mindspore.cn/



(a) Cover rate varying α (b) Cover rate varying β (c) Cover rate varying K (d) Cover rate varying w (e) Cover rate varying z

(f) Time varying α (g) Time varying β (h) Time varying K (i) Time varying w (j) Time varying z

(k) Space varying α (l) Space varying β (m) Space varying K (n) Space varying w (o) Space varying z

Fig. 4: Performance evaluation of RTK-Sketch.

TABLE I: Performance of LTR Model

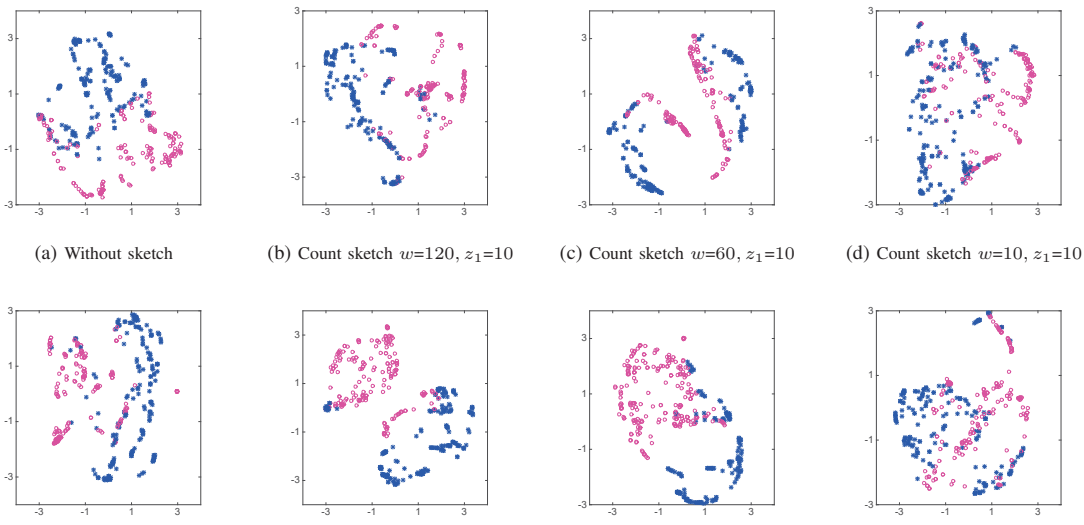
		ERR	nDCG@10	nDCG
Local	Party A	0.596	0.757	0.807
	Party B	0.620	0.746	0.807
	Party C	0.514	0.645	0.750
	Party D	0.554	0.754	0.796
	Average	0.571	0.725	0.790
Local+	Party A	0.609	0.763	0.811
	Party B	0.570	0.757	0.794
	Party C	0.538	0.713	0.781
	Party D	0.548	0.717	0.793
	Average	0.566	0.738	0.795
Global		0.555	0.738	0.790
CS-F-LTR		0.580	0.756	0.798

B. Evaluation of RTK-Sketch

Impact of Parameter α, β and K. The impact of α is shown in the first column of Fig. 4. We can find that with the increase of α the cover rate also increases fast. With a relatively large α, like $\alpha = 5$, the cover rate will approach to 1 and its growth will be slower. The results can guide us to choose an appropriate but not too large α, as the time and space costs will also increase linearly. The second column of Fig. 4 shows the impact of β. We observe that the cover rate decreases with larger β and setting $\beta \leq 0.2$ would be a reasonable choice. The time cost also decreases with larger β but the degree is not obvious while the space cost remains the same. The impact of K is in the third column of Fig. 4. We find that RTK-Sketch tends to have better approximation with larger K. And the time and space costs also increase linearly with K, which is consistent with our complexity analysis.

Impact of Sketch Size. The impact of sketch size w and z can be found in the last two columns of Fig. 4. About the cover rate, we find that larger size of sketch can perform better in finding the top-K relevant documents but there also exists exception when z increases from 50 to 70. The possible reason is that z controls the confidence of sketch results and when it is large enough the randomness will take control. We also compare RTK-Sketch with the NAIVE solution in time and space costs. We can find that the acceleration of RTK-Sketch is significant, from over 100 seconds to less than 10 ms even for a single query. The space cost also decreases roughly to $1/5$ of the NAIVE solution. The results verify both the effectiveness and efficiency of RTK-Sketch. We can conclude that by choosing some appropriate parameters, the utility loss brought by RTK-Sketch can be negligible.

Visualization of Sketch in LTR. We visualize different strategies of sketches and the results are shown in Fig. 5. We randomly sample 400 positive and negative instances in total, and apply different sketches to evaluate the influence. To show



(a) Without sketch (b) Count sketch w=120, z_1=10 (c) Count sketch w=60, z_1=10 (d) Count sketch w=10, z_1=10

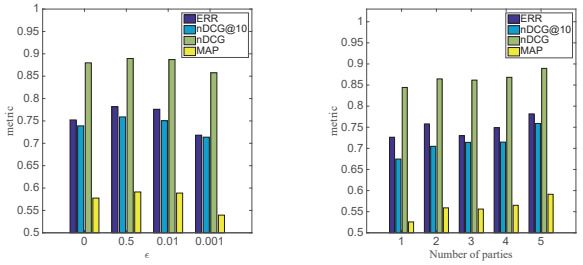
(e) CM sketch w=120, z_1=10 (f) Count sketch w=120, z_1=5 (g) Count sketch w=120, z_1=3 (h) Count sketch w=120, z_1=1

Fig. 5: Visualization of different sketch strategies. We sample in total 400 positive (red circles, score $= 1$ or 2) and negative (blue stars, score $= 0$) samples and embed them to 2D vectors.

the results more clearly, we embed the points into 2D plane with TSNE. Fig. 5a shows the results without sketch and Fig. 5b corresponds to the strategy in CS-F-LTR. We can see that the boundary is still discernible after applying Count sketch. We also try the CM sketch with same parameter setting and the results in Fig. 5e show similar performance to Count sketch. With the decrease of hash range w, we observe from Fig. 5c and 5d that the noise increases, indicating that the accuracy is sensitive to the hash range. With a smaller hash range, more terms will collide with each other, which results in inaccurate TF and other features. However, with the decrease of number of hash functions z_1, we find that the results are more robust. Even when $z_1 = 5$ (Fig. 5f) or $z_1 = 3$ (Fig. 5g), there is still a clear boundary. The boundary becomes unclear until z_1 decreases to 1 (Fig. 5h). It verifies that with a fixed z, a smaller z_1 can still produce accurate features while the privacy can be better preserved with more obfuscated terms. Overall, it shows that the utility loss brought by sketching algorithms to the conditional distribution in the classification task can be negligible with relatively large w and z_1.

C. Evaluation of CS-F-LTR

Main Results. In Table I, we record the results of Local, Local+, Global and CS-F-LTR with 3 evaluation metrics. We observe that CS-F-LTR outperforms Global and the averages of Local and Local+ on all the 3 measurements. However, we also find that it cannot always be beneficial to all the parties. As we can see from the table that party A and B have trained better local models than party C and D, which indicates that the data quality of A and B are higher. In this case, the improvement of performance for parties with low-quality data is significant. But for parties with high-quality data there is not always improvement, sometimes it even decreases the performance. It reveals a paradox from cross-silo federated



(a) Impact of privacy budget (b) Impact of number of parties

Fig. 6: Evaluation of privacy budget and number of parties.

learning, *i.e.*, when the data quality is highly biased among different parties, FL can be more beneficial to the parties with low-quality data while even be harmful to the parties with high-quality data. The challenge is unique in cross-silo settings as in a cross-device setting each mobile user only has a small amount of data (*i.e.*, low-quality data) and the global model can always be the best. How to address such fairness problem in FL remains an open question.

Impact of Privacy Budget. The impact of privacy budget is shown in Fig. 6a. We abuse $\epsilon = 0$ to represent the case that DP is not applied. We find surprisingly that with a small noise injected to features ($\epsilon = 0.5$), the performance can even be better. The possible reason is that the noise is added only to data with uncertain labels. Adding small noise can prevent the model from overfitting and improve the generalization ability. With the increase of noise, the performance starts to get worse, but it can still be controlled.

Impact of Number of Parties. The impact of number of parties can be found in Fig. 6b. We can see that nearly all

the evaluation metrics increase with larger number of parties. The nDCG@10 has an increase of 8% from single party to 5 parties, which verifies the effectiveness of CS-F-LTR. The increase of nDCG is not significant due to that there are many irrelevant documents for each query in test set. The ERR decreases at first, and increases by 5% finally. The possible reason is the imbalance of data which may influences some specific metrics when the number of parties is small.

D. Summary

We find that RTK-Sketch is both effective and efficient. It can decrease the querying time from over 100s to less than 10ms for a single query comparing with NAIVE solutions. We can also see that CS-F-LTR has advantages over horizontally FL (*i.e.*, Global) and local training (*i.e.*, the average of Local and Local+). Although it can bring large improvement for parties with low-quality data, it may be harmful to the parties with high-quality data, especially when the divergence of data quality among parties is large. The non-IID and fairness problems will remain open questions for FL in the future.

VII. Conclusion

In this paper, we study learning to rank (LTR) in a cross-silo federated learning (FL) setting and propose an FL framework, CS-F-LTR, which can help enterprises build specialized document retrieval systems collaboratively when each one only has limited data. To address the efficiency issues, we first propose a sketch and differential privacy based term frequency querying approach which has guarantees on both privacy and accuracy loss. Then we devise a new structure named RTK-Sketch which can significantly improve the overall efficiency of our algorithm. Finally, experiments on open dataset verify the efficiency and effectiveness of our solution.

Acknowledgment

We thank the anonymous reviewers for their valuable suggestions and comments. This work was partially supported by the National Key Research and Development Program of China under Grant No. 2018AAA0101100, the National Science Foundation of China (NSFC) under Grant No. 61822201 and U1811463, the CAAI-Huawei MindSpore Open Funding No. CAAIXSJLJJ-2020-020A, and the State Key Laboratory of Software Development Environment Open Funding No. SKLSDE-2020ZX-07. Yongxin Tong is the corresponding author in this paper.

References

[1] C. J. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. N. Hullender, "Learning to rank using gradient descent," in *ICML*, 2005, pp. 89–96.

[2] T. Liu, *Learning to Rank for Information Retrieval*. Springer, 2011.

[3] O. Chapelle and Y. Chang, "Yahoo! learning to rank challenge overview," in *ICML*, 2011, pp. 1–24.

[4] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM TIST*, vol. 10, no. 2, pp. 12:1–12:19, 2019.

[5] M. Barbaro, T. Zeller, and S. Hansell, "A face is exposed for aol searcher no. 4417749." *New York Times*, 2006.

[6] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *FOCS*, 1995, pp. 41–50.

[7] H. Pang, X. Ding, and X. Xiao, "Embellishing text search queries to protect user privacy," *PVLDB*, vol. 3, no. 1, pp. 598–607, 2010.

[8] M. Murugesan and C. Clifton, "Providing privacy through plausibly deniable search," in *SDM*, 2009, pp. 768–779.

[9] M. Gaboardi, E. J. G. Arias, J. Hsu, A. Roth, and Z. S. Wu, "Dual query: Practical private query release for high dimensional data," in *ICML*, 2014, pp. 1170–1178.

[10] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin, "Protecting data privacy in private information retrieval schemes," *J. Comput. Syst. Sci.*, vol. 60, no. 3, pp. 592–629, 2000.

[11] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," *Journal of Computer Security*, vol. 19, no. 5, pp. 895–934, 2011.

[12] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order-preserving encryption for numeric data," in *SIGMOD*, 2004, pp. 563–574.

[13] W. Zhang, Y. Lin, S. Xiao, J. Wu, and S. Zhou, "Privacy preserving ranked multi-keyword search for multiple data owners in cloud computing," *IEEE Trans. Computers*, vol. 65, no. 5, pp. 1566–1577, 2016.

[14] S. Ji, J. Shao, D. Agun, and T. Yang, "Privacy-aware ranking with tree ensembles on the cloud," in *SIGIR*, 2018, pp. 315–324.

[15] J. Konecný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *CoRR*, vol. abs/1610.05492, 2016.

[16] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, 2017, pp. 1273–1282.

[17] P. Kairouz, H. B. McMahan, B. Avent *et al.*, "Advances and open problems in federated learning," *CoRR*, vol. abs/1912.04977, 2019.

[18] K. Bonawitz, V. Ivanov, B. Kreuter *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *CCS*, 2017, pp. 1175–1191.

[19] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *ICLR*, 2018.

[20] D. Jiang, Y. Song, Y. Tong, X. Wu, W. Zhao, Q. Xu, and Q. Yang, "Federated topic modeling," in *CIKM*, 2019, pp. 1071–1080.

[21] Y. Wang, Y. Tong, and D. Shi, "Federated latent dirichlet allocation: A local differential privacy based framework," in *AAAI*, 2020, pp. 6283–6290.

[22] E. Kharitonov, "Federated online learning to rank with evolution strategies," in *WSDM*, 2019, pp. 249–257.

[23] S. Muthukrishnan, "Data streams: Algorithms and applications," *Foundations and Trends in Theoretical Computer Science*, vol. 1, no. 2, 2005.

[24] M. Charikar, K. C. Chen, and M. Farach-Colton, "Finding frequent items in data streams," *Theor. Comput. Sci.*, vol. 312, no. 1, pp. 3–15, 2004.

[25] G. Cormode and S. Muthukrishnan, "An improved data stream summary: the count-min sketch and its applications," *J. Algorithms*, vol. 55, no. 1, pp. 58–75, 2005.

[26] T. Yang, Y. Zhou, H. Jin, S. Chen, and X. Li, "Pyramid sketch: a sketch framework for frequency estimation of data streams," *VLDB*, vol. 10, no. 11, pp. 1442–1453, 2017.

[27] C. Masson, J. E. Rim, and H. K. Lee, "Ddsketch: A fast and fully-mergeable quantile sketch with relative-error guarantees," *VLDB*, vol. 12, no. 12, pp. 2195–2205, 2019.

[28] J. Li, Z. Li, Y. Xu, S. Jiang, T. Yang, B. Cui, Y. Dai, and G. Zhang, "Wavingsketch: An unbiased and generic sketch for finding top-k items in data streams," in *KDD*, 2020, pp. 1574–1584.

[29] J. Jiang, F. Fu, T. Yang, and B. Cui, "Sketchml: Accelerating distributed machine learning with data sketches," in *SIGMOD*, 2018, pp. 1269–1284.

[30] W. Zhu, P. Kairouz, H. Sun, B. McMahan, and W. Li, "Federated heavy hitters discovery with differential privacy," *CoRR*, vol. abs/1902.08534, 2019.

[31] C. Dwork, "Differential privacy," in *ICALP*, 2006, pp. 1–12.

[32] T. Li, Z. Liu, V. Sekar, and V. Smith, "Privacy for free: Communication-efficient learning with differential privacy using sketches," *CoRR*, vol. abs/1911.00972, 2019.

[33] S. E. Robertson, "Overview of the okapi projects," *Journal of Documentation*, vol. 53, no. 1, pp. 3–7, 1997.

[34] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *SIGIR*, 1998, pp. 275–281.

[35] G. Cormode and S. Muthukrishnan, "Summarizing and mining skewed data streams," in *SDM*, 2005, pp. 44–55.

[36] T. Qin and T. Liu, "Introducing LETOR 4.0 datasets," *CoRR*, vol. abs/1306.2597, 2013.