FedKC: Federated Knowledge Composition for Multilingual Natural Language Understanding

Haoyu Wang§, Handong Zhao†, Yaqing Wang§, Tong Yu†, Jiuxiang Gu†and Jing Gao§
§Purdue University, West Lafayette, Indiana, USA †Adobe Research, USA
§{wang5346, wang5075, jinggao}@purdue.edu, †{hazhao, tyu, jigu}@adobe.com

ABSTRACT

Multilingual natural language understanding, which aims to comprehend multilingual documents, is an important task. Existing efforts have been focusing on the analysis of centrally stored text data, but in real practice, multilingual data is usually distributed. Federated learning is a promising paradigm to solve this problem, which trains local models with decentralized data on local clients and aggregates local models on the central server to achieve a good global model. However, existing federated learning methods assume that data are independent and identically distributed (IID), and cannot handle multilingual data, that are usually non-IID with severely skewed distributions: First, multilingual data is stored on local client devices such that there are only monolingual or bilingual data stored on each client. This makes it difficult for local models to know the information of documents in other languages. Second, the distribution over different languages could be skewed. High resource language data is much more abundant than low resource language data. The model trained on such skewed data may focus more on high resource languages but fail to consider the key information of low resource languages. To solve the aforementioned challenges of multilingual federated NLU, we propose a plug-and-play knowledge composition (KC) module, called FedKC, which exchanges knowledge among clients without sharing raw data. Specifically, we propose an effective way to calculate a consistency loss defined based on the shared knowledge across clients, which enables models trained on different clients achieve similar predictions on similar data. Leveraging this consistency loss, joint training is thus conducted on distributed data respecting the privacy constraints. We also analyze the potential risk of FedKC and provide theoretical bound to show that it is difficult to recover data from the corrupted data. We conduct extensive experiments on three public multilingual datasets for three typical NLU tasks, including paraphrase identification, question answering matching, and news classification. The experiment results show that the proposed FedKC can outperform state-of-the-art baselines on the three datasets significantly.

CCS CONCEPTS

• **Computing methodologies → Distributed artificial intelligence**.

KEYWORDS

Federated learning, Multilingual natural language understanding

ACM Reference Format:
Haoyu Wang§, Handong Zhao†, Yaqing Wang§, Tong Yu†, Jiuxiang Gu†and Jing Gao§. 2022. FedKC: Federated Knowledge Composition for Multilingual Natural Language Understanding. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3485447.3511988

1 INTRODUCTION

Natural language understanding (NLU) [39] is one of the fundamental tasks in natural language processing (NLP). Used as an umbrella term, NLU refers to the efforts that make machines understand the context and meaning of natural languages, and thus it covers a variety of tasks including text classification, name entity recognition and sentiment analysis. Although NLU tasks have mainly been studied for documents written in a single language, the joint analysis of multilingual documents has attracted considerable attention as well. With the increasing information shared over Internet and mobile devices in a variety of languages, multilingual NLU is the key to connect billions of people across the globe.

Existing work on multilingual NLU has been focusing on the analysis of multilingual text data collected to a central server [34, 36]. In this paper, we study a practical and common scenario, in which multilingual text data is stored in distributed devices (clients). In many applications, it is difficult or even impossible to transmit all the data to central server due to privacy concerns. The objective is thus to comprehend multilingual documents without sharing raw data among clients.

To conduct multilingual NLU on distributed data, federated learning [3, 19, 20, 32, 58] could be adopted as a learning paradigm. In federated learning, a shared global model is trained under the coordination of a central server while keeping the user data decentralized on local clients. However, a straightforward adoption of federated learning paradigm for multilingual NLU does not work. Existing federated learning models assume that data are independent and identically distributed (IID), but this does not hold true in multilingual NLU. When data are non-IID, federated learning suffers from training instability due to its training on distributed data. [26, 63].

Here we use an example to illustrate the non-IID issue faced by many multilingual NLU tasks. Multilingual NLU is needed to understand the text stored on users' smart phones, and it is common that smart phone applications support multiple languages. Apple Siri and Amazon Alex support 21 and 8 languages respectively;

Google search supports 149 languages; and Instagram supports 36 languages. Users set their preferred languages (usually just one or two languages per user) for applications, and thus the data stored on each user's client is monolingual or bilingual. Therefore, the data among different clients are very different not only in styles and vocabularies but also in languages, leading to serious challenges for federated learning. Furthermore, the amount of data in different languages varies significantly. For example, the speakers of top three widely used languages account for 20% global populations[1]. Such a skewed distribution adds difficulty to the training of NLU models in a federated learning framework.

Federated learning algorithms, including FedAvg [31], and FedProx [27], are proposed to train a global model based on distributed data, but these methods cannot handle the aforementioned non-IID challenges that are observed in multilingual NLU tasks. Using FedAvg as an example, we explain how existing federated learning algorithm could fall short on non-IID data. In FedAvg, local model parameters are aggregated by weighted sum as the global model parameters, and weights are proportional to the amount of training data on clients. Therefore, the global model may be dominated by high resource languages and the information from low resource languages could be ignored. Also, the distribution among different languages could be skewed, and the distribution among different classes could also be imbalanced. FedAvg may suffer from poor performance on such non-IID data according to [27]. There are some efforts toward solving the non-IID issues in federated learning setting, which apply data augmentation via Mixup [60] or knowledge distillation [15]. These approaches are applied on image data. However, text data is a discrete sequence, which differs from pixel data of images. Therefore, Mixup strategy [41, 59, 60] is not suitable for the multilingual NLU task. As for knowledge distillation methods used in federated learning framework [24, 29, 35, 44], they usually need a shared auxiliary dataset. However, in multilingual NLU tasks, low resource language data is already scarce, so the requirement on extra auxiliary dataset cannot be met. All in all, existing federated learning algorithms may not address the challenges faced by multilingual NLU tasks.

In light of these challenges, we propose a federated knowledge composition mechanism (FedKC) for multilingual NLU tasks. FedKC takes advantage of federated learning framework to preserve users' privacy, and leverages knowledge composition to exchange knowledge (data embedding) among active clients. Specifically, we perform clustering on each client data to get the most representative knowledge, i.e. clustered data centroids, then exchange the learned data centroids among clients, breaking data island in federated learning to overcome non-IID and data imbalance challenges. The high-level learned knowledge is able to preserve data privacy, which is examined by both theoretical analyses and empirical studies in Section 5.6. The proposed consistency loss objective cannot be minimized directly because clients cannot access raw data in other clients. To tackle this challenge, we propose to conduct clustering on each client's data, use cluster centroids as the knowledge extracted from each client, and exchange the knowledge among clients. Because the centroids are the averaged data embeddings in each cluster, we name it knowledge composition (KC). The proposed KC is a plug-and-play module, which can be easily applied to multiple federated learning frameworks.

The contributions of the paper are summarized as following: 1) We propose a federated knowledge composition framework which is able to mitigate the non-IID challenge and protect privacy in multilingual natural language understanding tasks by exchanging only high-level knowledge among clients; 2) We propose a cluster-aware mechanism to reduce the approximation error in the knowledge exchange process; 3) We provide theoretical analysis for privacy guarantee in the knowledge transfer procedure; 4) We conduct extensive experiments on three benchmark datasets on multilingual natural language understanding, and the proposed method outperforms baselines significantly.

Table 1: The differences between FedKC and other federated learning methods. "model/data buffer storing" means storing last round global or local model, or sharing an auxiliary dataset. Both FedMD and FedED are knowledge distillation-based methods; both FedMix and XorMixFL extend mixup to federated learning frameworks.

Method	model/data buffer storing	mixed raw data sharing (w/o privacy guarantee)	avg. embedding sharing (w/ privacy guarantee)
FedAvg [31]	–	-	-
FedProx [27]	✓	-	-
MOON [25]	✓	-	-
FedMD [35]/FedED [44]	✓	-	-
FedMix [59]/XorMixFL [41]	-	✓	-
FedKC (ours)	-	-	✓

2 RELATED WORK

2.1 Multilingual Natural Language Understanding

Natural language understanding (NLU) contains several sub-tasks, e.g. paraphrase identification [21, 56, 57], natural language inference [30, 33, 37, 51], question answering [2, 5, 7, 55], news classification [6, 18], sentiment classification [45, 46, 52] and so on. With the increasing demand in multilingual environment, multilingual NLU has garnered extensive attention. Previous research on multilingual NLU is usually based on traditional deep learning models such as convolutional neural network (CNN) [23] and long short-term memory network (LSTM) [16] to learn multilingual word or sentence representation. For example, [47, 49, 50] learn multilingual word embedding; [53] leverages CNN to learn language-agnostic sentence representation; [54] applies a multi-task learning framework to learn character-based representation via CNN; [36] proposes hierarchical attention based on LSTM to learn multilingual document representation. However, with the development of NLP, recent works [1, 34, 38, 48] are mainly based on pre-trained multilingual language models such as mBERT [10], XLM [9], and XLM-RoBERTa [8]. They are pre-trained on large scale multilingual corpora, and then fine-tuned with few epochs to achieve outstanding performance in multiple NLU tasks. However, both traditional models and pre-trained models need centered data, which may raise data privacy issues in many real-world scenarios.

2.2 Federated Learning

To solve the potential privacy issues of machine learning models in practical use, federated learning has achieved more and more

[1]https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers

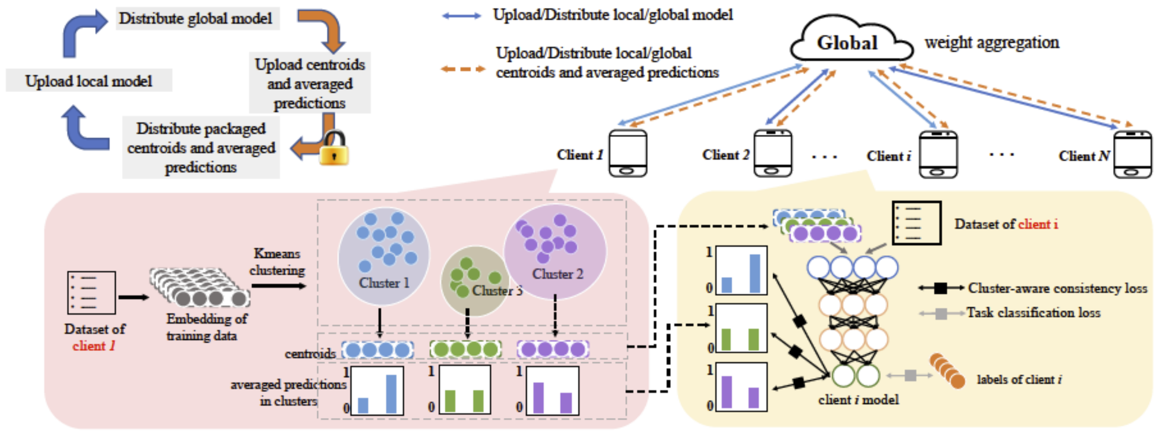
Distribute global model
Upload local model
Upload centroids and averaged predictions
Distribute packaged centroids and averaged predictions
Upload/Distribute local/global model
Upload/Distribute local/global centroids and averaged predictions
Global
weight aggregation
Client 1
Client 2
Client i
Client N
Dataset of client 1
Embedding of training data
Kmeans clustering
Cluster 1
Cluster 2
Cluster 3
centroids
averaged predictions in clusters
Dataset of client i
Cluster-aware consistency loss
Task classification loss
labels of client i
client i model

Figure 1: The framework of knowledge composition. The red background part clusters data, and computes the centroids and averaged predictions. The orange background part computes approximate consistency loss with distributed centroids and averaged predictions. Flow chart at top left corner shows the steps of uploading and distributing the model and data.

attention recently. Federated learning aims to learn a high-quality global with the aid of multiple local clients while forbidding data sharing among clients. FedAvg [31] is one of the most representative federated learning methods. It performs local stochastic gradient descent on local clients and the aggregates model parameters on server. However, FedAvg training sometimes is not stable [27]. Therefore, FedProx [27] adds regularizer to prevent parameters updating too far away from parameters in last communication round. To speedup FedAvg convergence, FedAdagrad [40], FedYogi [40], and FedAdam [40] are proposed. To handle the heterogeneity of local data distribution in clients, there is a lot of work based on contrastive learning, data augmentation and knowledge distillation. MOON [25] leverages contrastive learning to force parameters to be closer to global parameters in last round than local parameters in last round. FedMix [59] and XorMixFL [41] extend Mixup [60], one popular data augmentation method, to federated learning. However, it is difficult to apply mixup to achieve good performance on text data [4]. FEDDISTILL [64], FedMD [24], FedED [44] are three methods based on knowledge distillation. However, FedMD and FedED need an extra shared dataset to perform knowledge distillation, which is not available in many real-world cases. For FEDDISTILL, it needs to learn a generator. Although it is effective on images, it is not easy to be extended to text data, which is in a discrete space and hard to generate. Compared to these federated learning methods, the proposed FedKC does not need to hold a shared auxiliary dataset, and does not need to hold last round global or local model but only exchanges averaged embedding. We also summarize the difference between the proposed FedKC and other federated learning methods in Table 1.

3 BACKGROUND AND PRELIMINARIES

3.1 Problem Formulation

The federated learning setting includes a central server and N clients. The dataset on client i is denoted as $\mathcal{D}_i = \{(s_1^i, y_1^i), ..., (s_{n_i}^i, y_{n_i}^i)\}$, where s_j^i is the text content, y_j^i is the label of s_j^i, and n_i is the number of training instances in $\mathcal{D}_i$. The goal of federated learning is to learn a global model $F(s_i) \rightarrow y_i$ with decentralized data storage. In this paper, we focus on developing federated learning algorithms for multilingual NLU tasks, where multilingual dataset is denoted as $\{\mathcal{D}_i | i = 1, 2, ..., N\}$ with k classes.

3.2 Preliminaries

FedAvg. Federated averaging [31] is a popular and classic algorithm for federated learning. Within given communication round t, there are K active clients updating parameters locally. In federated learning, the central server first distributes global model parameters w^t to those active clients and then the active clients upload their updated parameters to the central server. After center server received updated parameters, the center server aggregates parameters to update the global model parameters via $w^{t+1} = \sum_i p_i w_i^{t+1}$, where client weight p_i is proportional to the amount of training data samples stored one client i, i.e. $p_i = \frac{n_i}{\sum_i n_i}$, and w_i^{t+1} is the updated parameters of client i in the t-th round.

4 METHODOLOGY

4.1 Overview

Federated learning includes two main procedures, where one is to update parameters on local clients and another is to aggregate client parameters for global model. Most widely adopted weight aggregation operation [31] is weighted sum by amount of training data on each client, thereby easily leading to model bias towards more emphasis on high resource languages. To overcome this issue, we propose a knowledge composition module to exchange knowledge among clients for federated learning. The knowledge composition involves two steps including knowledge sharing across clients (in Subsection 4.2) and updating client parameters via cluster-aware consistency loss (in Subsection 4.3). Our FedKC framework first conducts knowledge sharing across clients, then updates client parameters with cluster-aware consistency loss and defined task loss. In the end, we upload the updated client parameters into center server for weight aggregation via weighted average. With our proposed knowledge composition module, weighted average aggregation does not lead to model bias since each active client learns knowledge of low resource languages via client-aware consistency loss. Our framework is shown in Fig. 1.

4.2 Knowledge Sharing across Clients

For each client, the corresponding training data is usually not enough to cover entire task distribution to achieve a good model, especially in the multilingual scenario, where local clients may only have monolingual data. To alleviate this problem, we propose to exchange knowledge to bridge different clients. More specifically, we design consistency loss to force consistent predictions from different clients on representative data centroids. Formally, we optimize the following loss function for client i

$$\min_{w_i^t} - \sum_{j=1}^{n_i} \sum_{m=0}^{k-1} y_{jm}^i \log(F(s_j^i; w_i^t)[m]) + \alpha \sum_{m \in \mathcal{A}_t \setminus \{i\}} \sum_{k=1}^{n_m} CE(F(s_k^m; w_m^t), F(s_k^m; w_i^t)), \quad (1)$$

where α is the coefficient of knowledge distillation and $\mathcal{A}_t$ is the active client set at round t. The first term of Eqn. 1 is the classification loss and the second term of Eqn. 1 is the consistency loss.

However, the second term of Eqn. 1 needs to access raw data from other clients m ($m \in \mathcal{A}_t \setminus \{i\}$), which is against the privacy constraint in federated learning. Therefore, Eqn. 1 cannot be used as our optimized objective function. Furthermore, the input is a sequence consisting of discrete tokens, where arithmetic operations are difficult to perform for information privacy preserve. To target those issues, inspired by recent development of knowledge transfer [12, 13], we propose to transfer knowledge from the intermediate representation instead of the full network. Denote $f_p(\cdot;^p w_i^t)$ as the layers including p-th intermediate layer to the last layer and ${}^p e_j^i$ as the input embedding of the p-th intermediate layer respectively. For convenience, we omit notation p in following sections. Then we have the following new loss function

$$\min_{w_i^t} - \sum_{j=1}^{n_i} \sum_{m=0}^{k-1} y_{jm}^i \log(F(s_j^i; w_i^t)[m]) + \alpha \sum_{m \in \mathcal{A}_t \setminus \{i\}} \sum_{k=1}^{n_m} CE(f(e_k^m; w_m^t), f(e_k^m; w_i^t)), \quad (2)$$

According to [42], information may be leaked from embeddings via inversion attacks as well. Therefore, intermediate embeddings in Eqn. 2 still cannot provide privacy guarantee. To transfer knowledge without data leakage, we propose knowledge composition strategy, which uses corrupt embeddings alternatively to represent original embeddings approximately. Formally, we present the cross entropy with respect to client i and client m as

$$\mathcal{L} = \frac{1}{n_m} \sum_{k=1}^{n_m} CE(f(e_k^m; w_m^t), f(e_k^m; w_i^t)) = \frac{1}{n_m} \sum_{k=1}^{n_m} \ell(\hat{y}_k^m, e_k^m; w_i^t)).$$

Then we apply the first order taylor expansion to approximate $\ell(\hat{y}_k^m, e_k^m; w_i^t))$, which can be represented as

$$\ell(\hat{y}, e_k^m; w_i^t)) \approx \ell(\hat{y}, e; w_i^t)) + \frac{\partial \ell}{\partial x}(e_k^m - e) + \frac{\partial \ell}{\partial y}(\hat{y}_k^m - \hat{y}), \quad (3)$$

where $\frac{\partial \ell}{\partial x}$ and $\frac{\partial \ell}{\partial y}$ is evaluated at $x = e$ and $y = \hat{y}$. Therefore, we can rewrite the loss $\mathcal{L}$ in an approximate form:

$$\begin{aligned} \mathcal{L} =& \frac{1}{n_m} \sum_{k=1}^{n_m} (\ell(\hat{y}, e; w_i^t)) + \frac{\partial \ell}{\partial x}(e_k^m - e) + \frac{\partial \ell}{\partial y}(\hat{y}_k^m - \hat{y})) \\ =& \ell(\hat{y}, e; w_i^t)) + \frac{\partial \ell}{\partial x}(\frac{1}{n_m} \sum_{k=1}^{n_m} e_k^m - e) + \frac{\partial \ell}{\partial y}(\frac{1}{n_m} \sum_{k=1}^{n_m} \hat{y}_k^m - \hat{y}). \end{aligned} \quad (4)$$

Eqn. 4 indicates that it is not necessary to share the whole embeddings among clients since only corrupt embedding and predictions are needed. Then the next question becomes how to design e and $\hat{y}$. The choice of e and $\hat{y}$ will determine the approximation error considering that we use the first order taylor expansion to approximate consistency loss. Then we consider how to choose e and $\hat{y}$ to reduce the approximation error. Based on the error bound of taylor expansion [43], we have

$$\begin{aligned} &\|\ell(\hat{y}, e_k^m; w_i^t)) - \ell(\hat{y}, e; w_i^t)) + \frac{\partial \ell}{\partial x}(e_k^m - e) + \frac{\partial \ell}{\partial y}(\hat{y}_k^m - \hat{y})\|_2^2 \\ \leq& M(\|e_k^m - e\|_2^2 + \|\hat{y}_k^m - \hat{y}\|_2^2), \end{aligned} \quad (5)$$

where M is a constant. Therefore, we get the following optimization problem

$$\min_{e, \hat{y}} \frac{1}{n_m} \sum_{k=1}^{n_m} (\|e_k^m - e\|_2^2 + \|\hat{y}_k^m - \hat{y}\|_2^2). \quad (6)$$

Eqn. 6 can be solved to obtain $e = \frac{1}{n_m} \sum_{k=1}^{n_m} e_k^m, \hat{y} = \frac{1}{n_m} \sum_{k=1}^{n_m} \hat{y}_k^m$. The solution to Eqn. 6 shows that the corrupt embedding and prediction are the average of data embeddings and predictions. We present theoretic privacy guarantee regarding the proposed knowledge transfer option in Section 4.4. After applying the learned solution to Eqn. 3, we get a succinct and elegant approximation formulation:

$$\mathcal{L} = \ell(\frac{1}{n_m} \sum_{k=1}^{n_m} \hat{y}_k^m, \frac{1}{n_m} \sum_{k=1}^{n_m} e_k^m; w_i^t)). \quad (7)$$

4.3 Cluster-aware Consistency Loss

In Section 4.2, we propose to share knowledge across clients and use the consistency loss to bridge different clients. However, simply averaging embeddings and predictions but ignoring diverse patterns of embeedings leads to information loss and may result in high approximation errors. Thus, it is desired to transfer more fine-grained information and further minimize the approximation error.

To solve this problem, we propose a simple yet effective fine-grained cluster-aware consistency loss based on KMeans [14, 61], namely knowledge composition (KC). We apply KMeans to embeddings on each active client to get q clusters. Instead of using average embeddings, we propose to represent embeddings via several average embeddings of clusters (i.e., the centroid). The embeddings belonging to the same cluster usually carry similar patterns and are close to each other in term of Euclidean distance and thus using several centorids may capture more patterns and achieve smaller approximation error compared to one average embedding [62]. Such a statement is proven in the following proposition.

Proposition 1. *For embeddings e_k^m $(k = 1, ..., n_m)$, after KMeans clustering with q clusters, we have*

$$\sum_{k=1}^{n_m}(\|e_k^m - e\|_2^2 + \|\hat{y}_k^m - \hat{y}\|_2^2) \leq \sum_{i=1}^{q}\sum_{k\in C_i}(\|e_k^m - \mu_i\|_2^2 + \|\hat{y}_k^m - \hat{y}_i\|_2^2),$$

where μ_i is the i-th centroid, and $\hat{y}_i = \frac{1}{|C_i|}\sum_{k\in C_i}\hat{y}_k^m$.

The proof is provided in appendix.

In the following, we show the fine-grained knowledge composition in detail. Formally, we apply KMeans clustering to embeddings in client m

$$\min_{\mu}\sum_{i=1}^{q}\sum_{k=1}^{n_m} w_{ik}\|e_k^m - \mu_i\|_2^2, s.t. w_{ik} \text{ in } \{0,1\},$$

and we achieve q clusters C_i $(i = 1, 2, ..., q)$. For embeddings in each cluster, we apply taylor expansion on $e = \frac{1}{|C_i|}\sum_{k\in C_i} e_k^m$, $y = \hat{y}_i = \frac{1}{|C_i|}\sum_{k\in C_i}\hat{y}_k^m$ to approximate loss function in this cluster. In other words, we leverage q learned fine-grained data points to approximately represent all data points instead of simply averaging embeddings. The *cluster-aware consistency loss* can be written as

$$\mathcal{L} = \frac{1}{q}\sum_{i=1}^{q}\ell(\frac{1}{|C_i|}\sum_{k\in C_i} e_k^m, \frac{1}{|C_i|}\sum_{k\in C_i}\hat{y}_k^m; w_i^t)). \quad (8)$$

KMeans is usually used to cluster low dimensional features and may face some challenges while applying it to high dimensional embeddings [11]. For pre-trained multilingual language models like mBERT, its intermediate layer input embedding $e_k^m \in \mathbb{R}^{s\times d}$ is a matrix, where s is the number of tokens of the input text, and d is the dimension of token embedding. Therefore, when we use taylor expansion, we need to vectorize the matrix, e.g.Vec$(E_k^m) \in \mathbb{R}^{sd}$ firstly and then apply KMeans (We use E_k^m and e_k^m to distinguish vectors and matrices in the following.). Unfortunately, the vectorized embedding is in extremely high dimension. For example, for a text sequence with 128 tokens, the vectorized embedding is 98,304-dimensional using multilingual BERT. For such a high dimension, it is very time consuming to adopt KMeans directly. Therefore, we need to reduce dimension firstly and then use KMeans. However, what makes the problem thornier is that dimension reduction methods are also time consuming, such as PCA, auto-encoder, and thus cannot accelerate this procedure. To solve this problem, we propose a simple but effective method. The method is to perform KMeans based on low dimension embeddings to approximate intermediate layer KMeans. Formally, we have intermediate layer embeddings E_k^m, and the network defines a mapping $M(\text{Vec}(E_k^m)) \rightarrow e_k^m$, where e_k^m is a low dimensional embedding (For example, the pooling layer output of mBERT). Then we perform KMeans for e_k^m and get clusters C_i $(i = 1, 2, ..., q)$. The q fine-grained representative data points can be formulated as $e = \frac{1}{|C_i|}\sum_{k\in C_i}\text{Vec}(E_k^m)$, $y = \hat{y}_i = \frac{1}{|C_i|}\sum_{k\in C_i}\hat{y}_k^m$. By this way, we avoid performing KMeans in high dimension directly. We use proposition 2 to show that the proposed method can preserve clustering accuracy as well.

Proposition 2. *The distance between Vec(E_i^m) and Vec(E_j^m) can be bounded by the distance between e_i^m and e_j^m.*

The proof is provided in appendix.

4.4 Privacy Analysis and Communication Cost of Knowledge Composition

In this section, we provide privacy analysis of knowledge composition, present privacy guarantees for knowledge composition and analyze the additional communication cost brought by knowledge composition.

4.4.1 Privacy Issues of Knowledge Composition. Knowledge composition transfers the average of raw data embeddings and predictions among clients. Although it only shares extremely limited information of data, it still raises some privacy concerns. To address concerns, we provide a quantitative risk measurement regarding the attack and defense perspectives, where techniques are not presented in details since this is not a main focus of this paper.

According to [42], embeddings may leak raw data information when facing embedding inversion attack. However, in [42], this attack method can only target sentence embedding which can be considered as the aggregation of token embedding. Different with the setting in [42], we transfer the corrupt data embedding, making embedding inversion attack less effective. To the best of our knowledge, there is no successful attack claimed in this challenging setting. We try the attack method in [42] with a necessary Gaussian distribution assumption for embedding and show Proposition 3 as follows to elaborate that the attack results will be far away from the ground truth tokens.

Proposition 3. *Suppose $x_1, x_2, ..., x_n \sim \mathcal{N}(\mu, \sigma^2 I)$, and $\bar{x} = \frac{1}{k}\sum_{j=1}^{k} x_{i_j}$, where $i_1, i_2, ..., i_k \in \{1, 2, ..., n\}$, and $i_p \neq i_q$ for $p \neq q$. The optimal solution of the optimization problem $\min_Z ||f(V^T Z) - x_i||_2^2$ is Z_i^* and the optimal solution of the optimization problem $\min_Z ||f(V^T Z) - \bar{x}||_2^2$ is $\bar{Z}^*$. And the solutions gotten via projected gradient descent of the two optimization problems are Z_i and $\bar{Z}$ respectively. The residual errors are $\epsilon_i = f(V^T Z_i^*) - x_i$ and $\bar{\epsilon} = f(V^T\bar{Z}^*) - x_2$. Then with at least probability $1 - e^{-\delta}$, where $\delta > 0$, the following inequality holds:*

$$||Z_i^* - \bar{Z}||_F \geq (d - 2\sqrt{d\delta})\frac{k-1}{kL||V||_F}\sigma^2 - \frac{||\epsilon_i - \bar{\epsilon}||_2}{L||V||_F} - ||\bar{Z}^* - \bar{Z}||_F.$$

The proof is shown in appendix. Based on the proposition, we can find the distance between recovered data $\bar{Z}$ and true data Z_i^* is larger than $\frac{1}{L||V||_F}((d - 2\sqrt{d\delta})(1 - \frac{1}{k})\sigma^2 - ||\epsilon_i - \bar{\epsilon}||_2) - ||\bar{Z}^* - \bar{Z}||_F$.

4.4.2 Communication Cost of Knowledge Composition. Knowledge composition can be plugged into most popular federated learning framework such as FedAvg, FedProx, etc. with limited extra communication costs. The additional communication cost of knowledge composition is from that the corrupt data points need to be uploaded and distributed. We denote the number of active clients as n_a. Then uploading qn_a corrupt data points incurs $2qn_a(d + k)$ cost, where k is the number of label categories and d represents the embedding dimension. Consider a setting with 10 clusters, 10 activate clients and 10 label categories, it just needs to upload and distribute additional **0.16M** parameters when using pooling layer representation of mBERT. Compared to mBERT with **110M** parameters, the additional communication cost is quite small (**0.15%** additional cost).

5 EXPERIMENT

In this section, we evaluate the proposed knowledge composition with the goal of answering the following questions.

RQ1 How does Knowledge Composition perform as compared to state-of-the-art baselines?
RQ2 Is the proposed cluster-aware consistency loss effective to improve model performance?
RQ3 How does the performance change with respect to different parameters?
RQ4 Does Knowledge Composition reduce the data leakage risk?

5.1 Datasets and Experiment Settings

5.1.1 Datasets. We use three public benchmark datasets including PAWS-X [17], QAM [28] and NC [28], which correspond to paraphrase identification, question answering matching, and news classification respectively. Because usually they are used for cross-lingual tasks (training on English data and testing on other languages), they do not include training data for languages other than English. Therefore, we combine English training data with other languages validation data as the training set, and still use the English validation set as the validation dataset. We evaluate the final global model on all language testing dataset, and report their performance for each language respectively. We summarize the statistics of all datasets in Table 2.

Table 2: Statistics of datasets.

Dataset	# of languages	Task	|Train|	|Dev|	|Test|
PAWS-X	7	paraphrase identification	61,401	2,000	14,000
QAM	3	QA matching	120,000	10,000	30,000
NC	5	news classification	140,000	10,000	50,000

5.1.2 Baselines. We compare the proposed knowledge composition with following baselines: mBERT [10], FedAvg [31], FedProx [27], MOON [25], and FedMix [59]. Fine-tuning mBERT with all languages provides the ceiling performance for federated models, and we denote it as mBERT (all languages). We also show fine-tuning mBERT with only English data, which is denoted as mBERT (only en). Details about baselines are shown in the appendix.

5.1.3 Evaluation Metric and Implementation Details. Following [10, 17, 28, 56], we use Accuracy and F1 to evaluate paraphrase identification, use Accuracy to evaluate QA matching and news classification. For both the accuracy and F1, the higher, the better. We show the implementation details in the appendix.

5.2 Performance Comparison

In this section, we report the performance of baselines and the proposed knowledge composition in Table 3 to answer RQ1. Based on the table, we have following findings.

First, using multilingual training data provides more benefits for the three tasks than using monolingual training data. In Table 3, mBERT fine-tuned on multilingual data has much better performance than mBERT fine-tuned on only English data, e.g. 7.0%, 4.8% and 11.2% accuracy improvement corresponding to PI, QA and NC tasks respectively. It shows for multilingual NLU tasks, it is necessary for the model to learn from different languages. It also verifies our motivation which takes advantage of federated learning to learn multilingual model.

Second, baselines still have huge performance gap compared with data centered training. Compared to mBERT trained on centered multilingual data, federated learning models including FedAvg, FedProx, MOON and FedMix all have significant drops of accuracy. Because FedAvg simply weighted sums parameters of local clients, it is difficult to learn from clients with different languages. And it is clear that FedAvg performs worst in most cases. FedProx and MOON perform better than FedAvg in the three tasks. Both of them add regularizers based on FedAvg to update parameters smoothly to learn better representations. However, similar to FedAvg, they do not transfer knowledge among different clients as well. Therefore, it is still different for them to learn a good global model. For FedMix, it applies Mixup to federated learning. Mixup is a data augmentation method, which can exchange some knowledge among clients. However, Mixup is difficult to work on text data as good as on image data according to [4]. Therefore, it does not show good performance on all datasets.

Third, the proposed knowledge composition outperforms federated learning baselines greatly. We apply knowledge composition to two popular federated learning frameworks FedAvg and FedProx respectively. In Table 3, we find FedAvg+FedKC (all layers) performs better than FedAvg+FedKC (last layer). For FedAvg+FedKC (last layer), it only updates fine-tuning layers when performing knowledge composition, but for FedAvg+FedKC (all layers), it updates all transformer layers and fine-tuning layers when performing knowledge composition. Therefore, FedAvg+FedKC (all layers) can benefit more from knowledge composition and has better performance. Compared with FedProx+FedKC (all layers), FedAvg+FedKC (all layers) has similar performance. Although FedProx outperforms FedAvg, after using knowledge composition, the benefits brought by knowledge composition compensate for the defects of FedAvg. The regularizer used in FedProx just smooths the training process instead of providing additional information. So simply applying knowledge composition to FedAvg achieves good performance.

5.3 Cluster-aware Consistency Loss

In this section, we do the ablation study to answer RQ2. We report the results of knowledge composition with and without KMeans in Table 4. FedAvg+FedKC (last layer) and FedAvg+FedKC (all layers) represent knowledge composition with KMeans, and FedAvg+FedKC (last layer)\K and FedAvg+FedKC (all layers)\K represent knowledge composition without KMeans in the table.

According to the table, we find that knowledge composition with KMeans significantly outperforms knowledge composition without KMeans. We conduct the experiment on all the three datasets. FedAvg+FedKC (all layers) has averagely 2.0%, 2.9% improvement compared with FedAvg+FedKC (all layers)\K on PI and NC task respectively; FedAvg+FedKC (last layer) has averagely 2.6%, 2.6% improvement compared with FedAvg+FedKC (last layer)\K on PI and QA task respectively. It shows the effectiveness of leveraging KMeans to learn fine-grained approximation. This design brings significant improvements for both updating all transformer layers and fine-tuning layers.

Table 3: Comparison with baselines on the three datasets. "FedKC (last layer)" means performing knowledge composition for layers after the pooling layer of mBERT. "FedKC (all layers)" means performing knowledge composition for layers after the embedding layer of mBERT. The highest scores per category are bold.

Task	Method	en	de	es	fr	ru	ja	ko	zh	AVG
PI (Acc)	mBERT(only en)	0.9395	0.8515	0.8755	0.8705	-	0.7210	0.7035	0.7715	0.8190
	mBERT(all languages)	0.9515	0.9005	0.9025	0.9110	-	0.8165	0.8065	0.8435	0.8760
	FedAvg	0.7743	0.7253	0.7352	0.7336	-	0.6665	0.6551	0.6871	0.7110
	FedProx	0.8810	0.8191	0.8307	0.8306	-	0.7376	0.7253	0.7689	0.7990
	MOON	0.8798	0.8155	0.8279	0.8284	-	0.7342	0.7208	0.7618	0.7955
	FedMix	0.7836	0.7366	0.7501	0.7472	-	0.6772	0.6713	0.7032	0.7242
	FedAvg+FedKC(last layer)	0.8837	0.8239	0.8326	0.8367	-	0.7423	0.7262	0.7730	0.8026
	FedAvg+FedKC(all layers)	0.9032	0.8408	**0.8550**	**0.8557**	-	**0.7614**	0.7381	0.7890	**0.8204**
	FedProx+FedKC(all layers)	**0.9036**	**0.8416**	0.8486	0.8483	-	0.7578	**0.7457**	**0.7904**	0.8194
PI (F1)	mBERT(only en)	0.9341	0.8374	0.8610	0.8567	-	0.6437	0.5675	0.7326	0.7761
	mBERT(all languages)	0.9475	0.8900	0.8943	0.9032	-	0.7998	0.7886	0.8294	0.8647
	FedAvg	0.8026	0.7444	0.7640	0.7635	-	0.6825	0.6528	0.7175	0.7325
	FedProx	0.8699	0.8088	0.8194	0.8186	-	0.7154	0.6940	0.7548	0.7830
	MOON	0.8703	0.8036	0.8203	0.8193	-	0.7160	0.6954	0.7484	0.7819
	FedMix	0.8141	0.7602	0.7786	0.7739	-	0.6875	0.6767	0.7192	0.7443
	FedAvg+FedKC(last layer)	0.8704	0.8049	0.8177	0.8214	-	0.7295	0.6940	0.7608	0.7855
	FedAvg+FedKC(all layers)	0.8932	0.8258	**0.8403**	**0.8407**	-	0.7269	0.6926	0.7690	0.7984
	FedProx+FedKC(all layers)	**0.8939**	**0.8269**	0.8344	0.8341	-	**0.7313**	**0.7013**	**0.7727**	**0.7992**
QA	mBERT(only en)	0.6875	0.6436	-	0.6571	-	-	-	-	0.6627
	mBERT(all languages)	0.6929	0.6915	-	0.6992	-	-	-	-	0.6945
	FedAvg	0.6258	0.5649	-	0.6075	-	-	-	-	0.5994
	FedProx	0.6368	0.5611	-	0.6260	-	-	-	-	0.6080
	MOON	0.6307	0.6055	-	0.6461	-	-	-	-	0.6274
	FedMix	0.6374	0.6114	-	0.6464	-	-	-	-	0.6317
	FedAvg+FedKC(last layer)	0.6341	0.6154	-	0.6488	-	-	-	-	0.6328
	FedAvg+FedKC(all layers)	**0.6420**	**0.6285**	-	0.6360	-	-	-	-	0.6355
	FedProx+FedKC(all layers)	0.6401	0.6176	-	**0.6542**	-	-	-	-	**0.6373**
NC	mBERT(only en)	0.9157	0.7532	0.7556	0.7171	0.7242	-	-	-	0.7732
	mBERT(all languages)	0.9117	0.8698	0.8394	0.8128	0.8669	-	-	-	0.8601
	FedAvg	0.8806	0.7774	0.7185	0.7289	0.7305	-	-	-	0.7672
	FedProx	0.8788	0.7645	0.7216	0.7196	0.7391	-	-	-	0.7650
	MOON	0.8710	0.7192	0.6868	0.6970	0.7346	-	-	-	0.7417
	FedMix	0.8760	0.7111	0.6652	0.6900	0.7252	-	-	-	0.7335
	FedAvg+FedKC(last layer)	0.8814	0.7818	0.7385	0.7437	0.7463	-	-	-	0.7783
	FedAvg+FedKC(all layers)	**0.8836**	**0.7939**	**0.7454**	**0.7420**	**0.7553**	-	-	-	**0.7840**
	FedProx+FedKC(all layers)	0.8824	0.7858	0.7451	0.7333	0.7511	-	-	-	0.7795

Table 4: Comparison with knowledge composition with or without KMeans. "\K" means knowledge composition without KMeans. The highest scores per category are bold.

Task	Method	en	de	es	fr	ru	ja	ko	zh	AVG
PI (Acc)	FedAvg+FedKC(last layer)\K	0.8671	0.8007	0.8124	0.8082	-	0.7243	0.7121	0.7489	0.7820
	FedAvg+FedKC(all layers)\K	0.8882	0.8258	0.8370	0.8390	-	0.7400	0.7274	0.7753	0.8047
	FedAvg+FedKC(last layer)	0.8837	0.8239	0.8326	0.8367	-	0.7423	0.7263	0.7730	0.8026
	FedAvg+FedKC(all layers)	**0.9032**	**0.8408**	**0.8550**	**0.8557**	-	**0.7614**	**0.7381**	**0.7890**	**0.8204**
QA	FedAvg+FedKC(last layer)\K	0.6238	0.5907	-	0.6350	-	-	-	-	0.6165
	FedAvg+FedKC(all layers)\K	0.6381	**0.6184**	-	0.6443	-	-	-	-	0.6336
	FedAvg+FedKC(last layer)	0.6341	0.6154	-	0.6488	-	-	-	-	0.6328
	FedAvg+FedKC(all layers)	**0.6401**	0.6176	-	**0.6542**	-	-	-	-	**0.6373**
NC	FedAvg+FedKC(last layer)\K	**0.8848**	0.7760	0.7267	0.7276	0.7384	-	-	-	0.7707
	FedAvg+FedKC(all layers)\K	0.8815	0.7738	0.7037	0.7097	0.7401	-	-	-	0.7618
	FedAvg+FedKC(last layer)	0.8814	0.7818	0.7385	**0.7437**	0.7463	-	-	-	0.7783
	FedAvg+FedKC(all layers)	0.8836	**0.7939**	**0.7454**	0.7420	**0.7553**	-	-	-	**0.7840**

5.4 Sensitivity w.r.t. The Number of Epochs

In this section, we study how the local updating epochs, an important hyper-parameter in federated learning, influences the performance of FedAvg+FedKC (all layers). We conduct the experiment on NC datasets with three different epoch number 1,2, and 4, and show the results in Fig. 2.

According to Fig. 2, we find that locally updating 1 epoch can not achieve performance as good as 2 epochs and 4 epochs. When local parameters are updated in 1 epoch each round, FedAvg degenerates into FedSGD [31], and model is more likely to be underfitting. Compared with locally updating 4 epochs, locally updating 2 epochs performs better in most languages because updating 4 epochs may suffer from overfitting. This phenomenon is consistent with fine-tuning pre-trained language models with centered data.

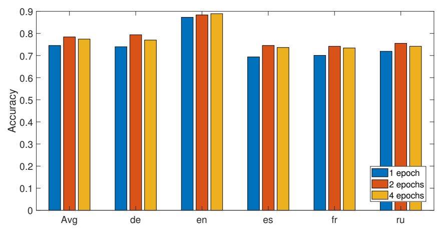

Figure 2: The results of FedAvg+FedKC (all layers) with different numbers of epochs in the local updating.

5.5 Sensitivity w.r.t. Active Clients

In this section, we study how the ratio of active clients influences the the performance of FedAvg+FedKC (all layers). We evaluate the performance of FedAvg+FedKC (all layers) with 2, 5, and 7 active clients each round on the NC dataset, and report results in Fig. 3.

From Fig. 3, we find that when the number of active clients is 2, it performs worse than 5 active clients and 7 active clients. Fewer active clients, less knowledge can be used to achieve a good global model. If the number of active clients is small and data distributions of these active clients are diverse, even though knowledge composition can exchange some knowledge among them, it is still very challenging to achieve high accuracy. However, when the number of active clients increases to 5 and 7, their performance is much better than that of 2 active clients, i.e. 11.9% and 15.0% improvement respectively. And we can find the global model will achieve better performance when there are more active clients, which is consistent with the phenomenon in FedAvg according to [31].

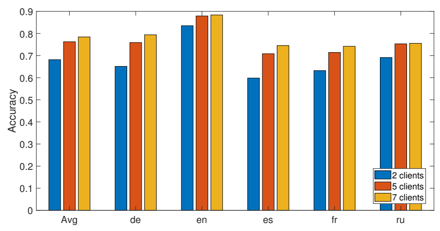

Figure 3: The results of FedAvg+FedKC (all layers) with different ratios of active clients.

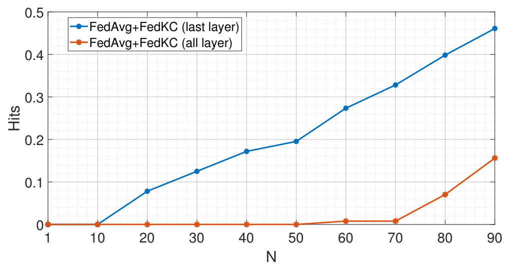

Figure 4: The hit ratio of recovered data. N represents the number of candidates and Hits means the hit ratio.

5.6 Qualitative and Quantitive Analysis of Privacy

In this section, we study the potential privacy risk of knowledge composition qualitatively and quantitively. We conduct the experiment on a subset of NC. Sentences used as the data in one cluster are shown in appendix due to limited space. We use the similar method in [42] to recover the raw text via averaged embedding. Inspired by the evaluation in information retrieval systems, we use Hits@N to evaluate the recovery accuracy. We set the max length of text as 128. For each token, we choose top-N possible candidates. Then the proportion of tokens who has candidates contained in the 10 raw text is the hit ratio, i.e., Hits. We show the result in Fig. 4.

According to Fig. 4, we find recovering data from averaged embeddings of both FedAvg+FedKC (last layer) and FedAvg+FedKC (all layers) can not achieve high accuracy. We show the Hits@N with varied N from 1 to 90. We can find the Hits@N value increases with the value of N. When the value of N is small, the recovered data is almost incorrect. And the Hits@90 of FedAvg+FedKC (last layer) is around 46.1% and the Hits@90 of FedAvg+FedKC (all layers) is only 15.6%. Therefore, recovering data from FedAvg+FedKC'(all layers) is more difficult than FedAvg+FedKC (last layer). The possible reason is the dimension of the vectorized representation of FedAvg+FedKC (all layers) is much higher than that of FedAvg+FedKC (last layer). Then we show the concrete cases of recovered tokens of FedAvg+FedKC (all layers). The recovered tokens include "into", "trip", "Russian", and "more". Except "Russian", it is difficult to achieve the useful information about the raw text from these recovered tokens. Therefore, it shows it is difficult to recover data from knowledge composition empirically.

6 CONCLUSION

In this paper, we propose a knowledge composition module which exchanges knowledge among clients to effectively handle non-IID challenges with privacy guarantee for multilingual NLU. Specifically, we perform clustering on each client's data to get the most representative knowledge, i.e. clustered data centroids, and exchange the learned data centroids among clients, breaking data island in federated learning to overcome non-IID and data imbalance challenges. The high-level knowledge can preserve data privacy, which is examined by both theoretical analyses and empirical studies. We conduct extensive experiments on three public multilingual NLU datasets including paraphrase identification, question answering matching, and news classification tasks. Experimental results show that the proposed knowledge composition outperforms state-of-the-art baselines on all three datasets.

ACKNOWLEDGMENTS

This work is supported in part by the US National Science Foundation under grant NSF IIS-1747614. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Sai Saketh Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. 2020. Deep learning models for multilingual hate speech detection. *arXiv preprint arXiv:2004.06465* (2020).

[2] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Learning to compose neural networks for question answering. *arXiv preprint arXiv:1601.01705* (2016).

[3] Priyam Basu, Tiasa Singha Roy, Rakshit Naidu, Zumrut Muftuoglu, Sahib Singh, and Fatemehsadat Mireshghallah. 2021. Benchmarking differential privacy and federated learning for bert models. *arXiv preprint arXiv:2106.13973* (2021).

[4] Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2021. A survey on data augmentation for text classification. *arXiv preprint arXiv:2107.03158* (2021).

[5] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075* (2015).

[6] Lia Bozarth and Ceren Budak. 2020. Toward a better performance evaluation framework for fake news classification. In *Proceedings of the international AAAI conference on web and social media*, Vol. 14. 60–71.

[7] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036* (2018).

[8] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116* (2019).

[9] Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems* 32 (2019), 7059–7069.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[11] Charles Elkan. 2003. Using the triangle inequality to accelerate k-means. In *Proceedings of the 20th international conference on Machine Learning (ICML-03)*. 147–153.

[12] Hao Fu, Shaojun Zhou, Qihong Yang, Junjie Tang, Guiquan Liu, Kaikui Liu, and Xiaolong Li. 2021. LRC-BERT: Latent-representation Contrastive Knowledge Distillation for Natural Language Understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 12830–12838.

[13] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision* 129, 6 (2021), 1789–1819.

[14] John A Hartigan and Manchek A Wong. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)* 28, 1 (1979), 100–108.

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[16] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[17] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*. PMLR, 4411–4421.

[18] Gurmeet Kaur and Karan Bajaj. 2016. News classification and its techniques: a review. *IOSR Journal of Computer Engineering (IOSR-JCE)* 18, 1 (2016), 22–26.

[19] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527* (2016).

[20] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492* (2016).

[21] Wuwei Lan and Wei Xu. 2018. Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. In *Proceedings of the 27th International Conference on Computational Linguistics*. 3890–3902.

[22] Beatrice Laurent and Pascal Massart. 2000. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics* (2000), 1302–1338.

[23] Yann LeCun, Yoshua Bengio, et al. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* 3361, 10 (1995), 1995.

[24] Daliang Li and Junpu Wang. 2019. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581* (2019).

[25] Qinbin Li, Bingsheng He, and Dawn Song. 2021. Model-Contrastive Federated Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10713–10722.

[26] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine* 37, 3 (2020), 50–60.

[27] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2018. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127* (2018).

[28] Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, et al. 2020. Xglue: A new benchmark dataset for cross-lingual pre-training, understanding and generation. *arXiv preprint arXiv:2004.01401* (2020).

[29] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. 2020. Ensemble distillation for robust model fusion in federated learning. *arXiv preprint arXiv:2006.07242* (2020).

[30] Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. 2016. Learning natural language inference using bidirectional LSTM model and inner-attention. *arXiv preprint arXiv:1605.09090* (2016).

[31] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[32] H Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. 2016. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629* (2016).

[33] Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2015. Natural language inference by tree-based convolution and heuristic matching. *arXiv preprint arXiv:1512.08422* (2015).

[34] Stephen Mutuvi, Emanuela Boros, Antoine Doucet, Gaël Lejeune, Adam Jatowt, and Moses Odeo. 2020. Multilingual Epidemiological Text Classification: A Comparative Study. In *COLING, International Conference on Computational Linguistics*.

[35] Wanning Pan and Lichao Sun. 2021. Global knowledge distillation in federated learning. *arXiv preprint arXiv:2107.00051* (2021).

[36] Nikolaos Pappas and Andrei Popescu-Belis. 2017. Multilingual hierarchical attention networks for document classification. *arXiv preprint arXiv:1707.00896* (2017).

[37] Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933* (2016).

[38] Sara Piscitelli, Edoardo Arnaudo, and Claudio Rossi. 2021. Multilingual Text Classification from Twitter during Emergencies. In *2021 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 1–6.

[39] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences* (2020), 1–26.

[40] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. 2020. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295* (2020).

[41] MyungJae Shin, Chihoon Hwang, Joongheon Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. 2020. Xor mixup: Privacy-preserving data augmentation for one-shot federated learning. *arXiv preprint arXiv:2006.05148* (2020).

[42] Congzheng Song and Ananth Raghunathan. 2020. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 377–390.

[43] Karl R Stromberg. 2015. *An introduction to classical real analysis*. Vol. 376. American Mathematical Soc.

[44] Dianbo Sui, Yubo Chen, Jun Zhao, Yantao Jia, Yuantao Xie, and Weijian Sun. 2020. Feded: Federated learning via ensemble distillation for medical relation extraction. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*. 2118–2128.

[45] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2015. Effective LSTMs for target-dependent sentiment classification. *arXiv preprint arXiv:1512.01100* (2015).

[46] Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*. 1422–1432.

[47] Shyam Upadhyay, Manaal Faruqui, Chris Dyer, and Dan Roth. 2016. Cross-lingual models of word embeddings: An empirical comparison. *arXiv preprint arXiv:1604.00425* (2016).

[48] Niels van der Heijden, Helen Yannakoudakis, Pushkar Mishra, and Ekaterina Shutova. 2021. Multilingual and cross-lingual document classification: A meta-learning approach. *arXiv preprint arXiv:2101.11302* (2021).

[49] Ivan Vulić, Douwe Kiela, Stephen Clark, and Marie Francine Moens. 2016. Multi-modal representations for improved bilingual lexicon learning. In *Proceedings of*

the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 188–194.

[50] Yogarshi Vyas and Marine Carpuat. 2016. Sparse bilingual word representations for cross-lingual lexical entailment. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*. 1187–1197.

[51] Shuohang Wang and Jing Jiang. 2015. Learning natural language inference with LSTM. *arXiv preprint arXiv:1512.08849* (2015).

[52] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*. 606–615.

[53] Joonatas Wehrmann, Willian Becker, Henry EL Cagnini, and Rodrigo C Barros. 2017. A character-based convolutional neural network for language-agnostic Twitter sentiment analysis. In *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2384–2391.

[54] Jônatas Wehrmann, Willian E Becker, and Rodrigo C Barros. 2018. A multi-task neural network for multilingual sentiment classification and language detection on twitter. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. 1805–1812.

[55] Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604* (2016).

[56] Wenpeng Yin and Hinrich Schütze. 2015. Convolutional neural network for paraphrase identification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 901–911.

[57] Wenpeng Yin and Hinrich Schütze. 2016. Discriminative phrase embedding for paraphrase identification. *arXiv preprint arXiv:1604.00503* (2016).

[58] Xuefei Yin, Yanming Zhu, and Jiankun Hu. 2021. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM Computing Surveys (CSUR)* 54, 6 (2021), 1–36.

[59] Tehrim Yoon, Sumin Shin, Sung Ju Hwang, and Eunho Yang. 2021. Fedmix: Approximation of mixup under mean augmented federated learning. *arXiv preprint arXiv:2107.00233* (2021).

[60] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017).

[61] Handong Zhao, Zhengming Ding, and Yun Fu. 2017. Multi-View Clustering via Deep Matrix Factorization. In *AAAI*. 2921–2927.

[62] Handong Zhao, Hongfu Liu, and Yun Fu. 2016. Incomplete Multi-Modal Visual Data Grouping. In *IJCAI*. 2392–2398.

[63] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. 2021. Federated learning on non-IID data: A survey. *Neurocomputing* 465 (2021), 371–390.

[64] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. 2021. Data-Free Knowledge Distillation for Heterogeneous Federated Learning. *arXiv preprint arXiv:2105.10056* (2021).

A BASELINES

We compare the proposed knowledge composition with following baselines:

- mBERT [10], which is a multilingual version of BERT, is a state-of-the-art model for multiple multilingual NLP tasks. It is trained on top 104 languages of Wikipedia corpora. Fine-tuning mBERT with all languages provides the ceiling performance for federated models, and we denote it as mBERT (all languages). We also show fine-tuning mBERT with only English data, which is denoted as mBERT (only en).
- FedAvg [31] is one of the most popular federated learning methods.
- FedProx [27], an improved version of FedAvg, adds a regularizer to prevent local parameters updating too much.
- MOON [25] introduces contrastive learning to force current local representation to be closer to global representation than last round local representation.
- FedMix [59] extends the data augmentation method Mixup to federated learning. It shows great performance on image data.

B IMPLEMENTATION DETAILS

For PAWS-X, there are 22 clients; for QAM, there are 20 clients; for NC, there are 70 clients. For each communication round, we set random 10% of clients as active clients, and local model updates 2 epochs. We set the whole communication T as 35. For PAWS-X and QAM, we set the number of clusters in KMeans as 10 and for NC, the number of clusters is set 5. We use the default setting of baselines for other parameters following [25, 27, 31, 59]. For all experiments, we run ten times and report the average results.

C PROOF OF PROPOSITION 1

Proposition 1. *For embeddings e_k^m ($k = 1, 2, ..., n_m$), after KMeans clustering with q clusters, we have*

$$\sum_{k=1}^{n_m}(\|e_k^m - e\|_2^2 + \|\hat{y}_k^m - \hat{y}\|_2^2) \leq \sum_{i=1}^{q}\sum_{k\in C_i}(\|e_k^m - \mu_i\|_2^2 + \|\hat{y}_k^m - \hat{y}_i\|_2^2), \tag{9}$$

where μ_i is the i-th centroid, and $\hat{y}_i = \frac{1}{|C_i|}\sum_{k\in C_i}\hat{y}_k^m$.

Proof. We can rewrite the left of the inequality as

$$\sum_{k=1}^{n_m}\|e_k^m - e\|_2^2 = \sum_{i=1}^{q}\sum_{k\in C_i}\|e_k^m - e\|_2^2. \tag{10}$$

Because KMeans is to learn the optimal centroids to minimize the reconstruction error, we have

$$\sum_{k\in C_i}\|e_k^m - e\|_2^2 \leq \sum_{k\in C_i}\|e_k^m - \mu_i\|_2^2. \tag{11}$$

Therefore, we can get the following inequality

$$\sum_{k=1}^{n_m}\|e_k^m - e\|_2^2 \leq \sum_{i=1}^{q}\sum_{k\in C_i}\|e_k^m - \mu_i\|_2^2. \tag{12}$$

Similarly, we also have

$$\sum_{k=1}^{n_m}\|\hat{y}_k^m - \hat{y}\|_2^2 = \sum_{i=1}^{q}\sum_{k\in C_i}\|\hat{y}_k^m - \hat{y}\|_2^2. \tag{13}$$

Because $\arg\min_{\hat{y}}\sum_{k\in C_i}\|\hat{y}_k^m - \hat{y}\|_2^2 = \frac{1}{|C_i|}\sum_{k\in C_i}\hat{y}_k^m$, we have

$$\sum_{k=1}^{n_m}\|\hat{y}_k^m - \hat{y}\|_2^2 \leq \sum_{i=1}^{q}\sum_{k\in C_i}\|\hat{y}_k^m - \hat{y}_i\|_2^2. \tag{14}$$

Therefore,

$$\sum_{k=1}^{n_m}(\|e_k^m - e\|_2^2 + \|\hat{y}_k^m - \hat{y}\|_2^2) \leq \sum_{i=1}^{q}\sum_{k\in C_i}(\|e_k^m - \mu_i\|_2^2 + \|\hat{y}_k^m - \hat{y}_i\|_2^2)$$

holds. □

D PROOF OF PROPOSITION 2

Proposition 2. *The distance between $Vec(E_i^m)$ and $Vec(E_j^m)$ can be bounded by the distance between e_i^m and e_j^m.*

Proof. The network defines a mapping $M(\text{Vec}(E_k^m)) \to e_k^m$. Therefore, using one order Taylor expansion, we have $e_k^m \approx M(\mathbf{0}) + (\frac{\partial M}{\partial \text{Vec}(X)}|_{\text{Vec}(X)=\mathbf{0}})^T\text{Vec}(E_k^m) = b + W^T\text{Vec}(\text{Vec}(E_k^m))$. Therefore, we have

$$\begin{aligned}\|e_i^m - e_j^m\|_2^2 &\approx \|W^T(\text{Vec}(E_i^m) - \text{Vec}(E_j^m))\|_2^2 \\ &= (\text{Vec}(E_i^m) - \text{Vec}(E_j^m))^T W W^T(\text{Vec}(E_i^m) - \text{Vec}(E_j^m)).\end{aligned} \tag{15}$$

Therefore, we have

$$\begin{aligned}\frac{1}{\lambda_{max}(WW^T)}\|e_i^m - e_j^m\|_2^2 \leq &\|\text{Vec}(E_i^m) - \text{Vec}(E_j^m)\|_2^2 \\ &\leq \frac{1}{\lambda_{min}(WW^T)}\|e_i^m - e_j^m\|_2^2,\end{aligned} \tag{16}$$

where $\lambda_{max}(WW^T)$ and $\lambda_{min}(WW^T)$ mean the largest and smallest eigenvalues of WW^T respectively. □

E PROOF OF PROPOSITION 3

Lemma 1. *Let $x \sim \chi_d^2$, then for all $\delta > 0$,*

$$P(x \geq d + 2\sqrt{d\delta} + 2\delta) \leq e^{-\delta}, \tag{17}$$

$$P(x \leq d - 2\sqrt{d\delta}) \leq e^{-\delta}. \tag{18}$$

Lemma 2. *Suppose $x_1, x_2, ..., x_n \sim \mathcal{N}(\mu, \sigma^2 I)$, and $\bar{x} = \frac{1}{k}\sum_{j=1}^{k}x_{i_j}$, where $i_1, i_2, ..., i_k \in \{1, 2, ..., n\}$, and $i_p \neq i_q$ for $p \neq q$. Then with at least probability $1 - e^{-\delta}$, where $\delta > 0$, the following inequality holds:*

$$||\bar{x} - x_i||_2^2 \geq (d - 2\sqrt{d\delta})(1 - \frac{1}{k})\sigma^2. \tag{19}$$

Proof. case 1: $x_i \notin \{x_{i_1}, ..., x_{i_k}\}$.

Consider random variable $y = \bar{x} - x_i$. We have $\mathbb{E}[y] = 0$, and $Var[y] = (1 + \frac{1}{k})\sigma^2 I$. Therefore, we have $y \sim \mathcal{N}(0, (1 + \frac{1}{k})\sigma^2 I)$. Then define a random variable $z = \frac{1}{\sqrt{1+\frac{1}{k}}\sigma}y$, and we have $z \sim \mathcal{N}(0, I)$. Therefore, we have $||z||_2^2 \sim \chi_d^2$, where d is the dimension of z. According to [22], for $\delta > 0$, we have

$$P(||z||_2^2 - d \leq -2\sqrt{d\delta}) \leq e^{-\delta}. \tag{20}$$

Therefore,

$$P(||z||_2^2 \geq d - 2\sqrt{d\delta}) \geq 1 - e^{-\delta}. \tag{21}$$

Then we have

$$P(||z||_2^2 \geq (d-2\sqrt{d\delta})(1+\frac{1}{k})\sigma^2) \geq 1-e^{-\delta}. \tag{22}$$

case 2: $x_i \in \{x_{i_1}, ..., x_{i_k}\}$.

Consider random variable $y = \bar{x} - x_i$. Without loss of generality, let x_i be x_{i_1} here. Therefore, $y = (\frac{1}{k}-1)x_{i_1} + \frac{1}{k}x_{i_2} + ... + \frac{1}{k}x_{i_k}$. We have $\mathbb{E}[y] = 0$, and $Var[y] = ((\frac{1}{k}-1)^2 + (\frac{1}{k})^2 k)\sigma^2 I = (1-\frac{1}{k})\sigma^2 I$. Therefore, we have $y \sim \mathcal{N}(0, (1-\frac{1}{k})\sigma^2 I)$. Then define a random variable $z = \frac{1}{\sqrt{1-\frac{1}{k}}\sigma} y$, and we have $z \sim \mathcal{N}(0, I)$. Therefore, we have $||z||_2^2 \sim \chi_d^2$. Similar to case 1, we have

$$P(||z||_2^2 \geq (d-2\sqrt{d\delta})(1-\frac{1}{k})\sigma^2) \geq 1-e^{-\delta}. \tag{23}$$

Because $(d-2\sqrt{d\delta})(1-\frac{1}{k})\sigma^2 < (d-2\sqrt{d\delta})(1+\frac{1}{k})\sigma^2$, with at least probability $1-e^{-\delta}$, where $\delta > 0$, the following inequality holds:

$$||\bar{x} - x_i||_2^2 \geq (d-2\sqrt{d\delta})(1-\frac{1}{k})\sigma^2. \tag{24}$$

□

Lemma 3. *The optimal solution with softmax of the optimization problem* $\min_Z ||f(V^T Softmax(Z)) - x_1||_2^2 + \lambda||Z||_1$ *is* Z_1^* *and the optimal solution of the optimization problem* $\min_Z ||f(V^T Softmax(Z)) - x_2||_2^2 + \lambda||Z||_1$ *is* Z_2^*. *And the solutions after softmax gotten via SGD of the two optimization problems are* Z_1 *and* Z_2 *respectively, where* $f(\cdot)$ *is a encoding function and* $f(\cdot)$ *is L-Lipschitz continuous. The residual errors are* $\epsilon_1 = f(V^T Z_1^*) - x_1$ *and* $\epsilon_2 = f(V^T Z_2^*) - x_2$. *We have*

$$\begin{aligned}||Z_1^* - Z_2||_F \geq &\frac{1}{L||V||_F}(||x_1 - x_2||_2 - ||\epsilon_1 - \epsilon_2||_2) \\ &- ||Z_2^* - Z_2||_F.\end{aligned} \tag{25}$$

Proof. We can rewrite x_1 and x_2 as

$$x_1 = f(V^T Z_1^*) - \epsilon_1, \tag{26}$$

$$x_2 = f(V^T Z_2^*) - \epsilon_2. \tag{27}$$

Therefore, we have

$$\begin{aligned}||x_1 - x_2||_2 =& ||f(V^T Z_1^*) - \epsilon_1 - f(V^T Z_2^*) + \epsilon_2||_2 \\ \leq& L||V||_F ||Z_1^* - Z_2^*||_F + ||\epsilon_1 - \epsilon_2||_2 \\ =& L||V||_F ||Z_1^* - Z_2 + Z_2 - Z_2^*||_2 + ||\epsilon_1 - \epsilon_2||_2 \\ \leq& L||V||_F ||(||Z_1^* - Z_2||_F + ||Z_2 - Z_2^*||_F) \\ &+ ||\epsilon_1 - \epsilon_2||_2\end{aligned} \tag{28}$$

Transpose and we can get

$$||Z_1^* - Z_2||_2 \geq \frac{1}{||V||_F}(||x_1 - x_2||_2 - ||\epsilon_1 - \epsilon_2||_2) - ||Z_2^* - Z_2||_2.$$

Here, $||Z_2^* - Z_2||_2$ can be bounded by the upper error bounds of the two optimization problem via projected gradient descent. □

Proposition 3. *Suppose* $x_1, x_2, ..., x_n \sim \mathcal{N}(\mu, \sigma^2 I)$, *and* $\bar{x} = \frac{1}{k}\sum_{j=1}^k x_{i_j}$, *where* $i_1, i_2, ..., i_k \in \{1, 2, ..., n\}$, *and* $i_p \neq i_q$ *for* $p \neq q$. *The optimal solution of the optimization problem* $\min_Z ||f(V^T Z) - x_i||_2^2$ *is* Z_i^* *and the optimal solution of the optimization problem* $\min_Z ||f(V^T Z) - \bar{x}||_2^2$ *is* $\bar{Z}^*$. *And the solutions gotten via projected gradient descent of the two optimization problems are* Z_i *and* $\bar{Z}$ *respectively. The residual errors are* $\epsilon_i = f(V^T Z_i^*) - x_i$ *and* $\bar{\epsilon} = f(V^T \bar{Z}^*) - x_2$. *Then with at least probability* $1 - e^{-\delta}$, *where* $\delta > 0$, *the following inequality holds:*

$$\begin{aligned}||Z_i^* - \bar{Z}||_F \geq &\frac{1}{L||V||_F}((d-2\sqrt{d\delta})(1-\frac{1}{k})\sigma^2 - ||\epsilon_i - \bar{\epsilon}||_2) \\ &- ||\bar{Z}^* - \bar{Z}||_F.\end{aligned} \tag{29}$$

Proof. According to Lemma 3, we have

$$\begin{aligned}||Z_i^* - \bar{Z}||_F \geq &\frac{1}{L||V||_F}(||x_i - \bar{x}||_2 - ||\epsilon_i - \bar{\epsilon}||_2) \\ &- ||\bar{Z}^* - \bar{Z}||_F.\end{aligned} \tag{30}$$

Then based on Lemma 2, with at least probability $1 - e^{-\delta}$, where $\delta > 0$, the following inequality holds:

$$\begin{aligned}||Z_i^* - \bar{Z}||_F \geq &\frac{1}{L||V||_F}((d-2\sqrt{d\delta})(1-\frac{1}{k})\sigma^2 - ||\epsilon_i - \bar{\epsilon}||_2) \\ &- ||\bar{Z}^* - \bar{Z}||_F.\end{aligned} \tag{31}$$

□

F TEXT USED IN SECTION 5.6

1) Kim Kardashian's baby bump is starting to make an appearance and the reality star is showing it off in tight, sexy, plunging outfits.
2) Sophia Tabers explains her stroke of good luck.
3) There's no one more synonymous with Old Hollywood glamour than Liz Taylor, but there's something special about seeing the icon when all the glitz and diamonds are stripped away.
4) Hundreds of people are missing and an unknown number believed dead after a partly-constructed hydropower dam in southeast Laos collapsed, sending flash floods surging through six villages.
5) It looks like the Kardashians will be attending Coachella in luxury this year. The famous reality TV family just purchased a $12 million mansion located in La Quinta's Madison Club near the popular music festival.
6) A Florida woman was arrested for driving under the influence of alcohol after users on the live video streaming app 'Periscope' called police warning she was intoxicated. (Oct. 13)
7) A recent Gallup poll asked Americans what they considered the greatest threats to the U.S. Do those numbers align with the government's perception?
8) """Federal prosecutors are accusing a 29-year-old woman of being a Russian spy. They say Maria Butina allegedly """"took steps to develop relationships with American politicians."""" CBS News Washington correspondent Paula Reid reports."""
9) At the Democratic debate, candidates were asked which enemies they're most proud of making.
10) The owner of Bar Marco in Pittsburg put an end to tipping his waiters in favor of paying his employees a 'living wage.' Not only are they getting more money, they are also getting: health insurance, paid time off, and equity in the company. Keri Lumm (@thekerilumm) reports.