Federated Learning for NLP

M1 Supervised Project

Part 1: Bibliographic Report

Submitted in partial fulfillment
of the requirements for the degree of
MSc Natural Language Processing

by

Anna MOSOLOVA
Elisa LUBRINI
Pierrick FOULQUIER

Supervisor:
Christophe CERISARA

Reviewer:
Marianne CLAUSEL

UNIVERSITÉ DE LORRAINE

Institut des sciences du Digital Management & Cognition
COMPOSANTE DE L'UNIVERSITÉ DE LORRAINE

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Academic year 2020-2021

Contents

Chapter 1

Introduction

This report represents a *literature review* and *research proposal* for the 2020-2021 NLP supervised project. After a brief introduction to the motivations behind our research interest, Chapter 2 will give a theoretical overview of the techniques that we are going to investigate and Chapter 3 will define the benchmarks to be used during the evaluation of our project. Finally, Chapter 4 will synthesise our research plans and expected milestones.

1.1 Data confidentiality: socio-political context

In May 2016, a data protection package was adopted by the European Union aiming at making its countries ”fit for the digital age.”[1]. When the Data Protection Regulation (GDPR) became active, an European Data Protection Board (EDPB) was established with the aim of ensuring the consistent application of data protection rules throughout the European Union.

Worries about the handling of private data seem to have now become a growing trend, not only in the EU, but worldwide. According to the United Nations, to this day, 132 out of 194 countries have produced some kind of legislation to secure, to various extents, data protection and privacy, with the importance of related issues being nowadays increasingly recognised[2].

In spite of the increasing difficulty in accessing data stored on personal devices, most of the companies still run machine learning models on a centralized server, which requires them to retrieve raw user data before using it for learning. Because of the complexity of issues related to data confidentiality, however, some companies ended up being fined for unethical practises (see, for example, some recent cases involving Google[3] and Facebook[4]). These events called once again the attention to the need to find a learning solution that can both grant user data confidentiality and allow companies to access the data they need to continue providing their services.

Additionally, uses of data for purposes that are commonly considered as ethi-

[1] https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en

[2] https://unctad.org/page/data-protection-and-privacy-legislation-worldwide

[3] https://ec.europa.eu/commission/presscorner/detail/en/IP_19_1770

[4] https://www.nytimes.com/2019/07/12/technology/facebook-ftc-fine.html

cal, such as research or crime prevention, would also benefit from easier access to larger datasets, meaning that finding a trade-off between confidentiality and data access would not only favour private companies, but also positively affect individual communities and, potentially, society as a whole, to some extent.

1.1.1 Federated learning: a possible solution?

Seemingly offering a solution to this problem, federated learning is a technique that has been raising in popularity since it was first introduced by Google in 2016 (§ Section 2.1.3). The concept behind federated learning (widely discussed in Chapter 2) is that of training models on users' devices so that service providers can benefit from a model that has full access to private data, without the need for them to access such data directly.

However, as explained in Section 2.1, federated learning is far from immune to attacks and it has its own set of limitations. Attacks, specifically, have the potential to undermine the confidentiality of the data which motivates the use of federated learning in the first place.

1.2 Purpose statement

The purpose of this report is recording our exploration of different techniques that could be used in conjunction with a federated learning protocol (i.e. ensemble learning, transfer learning and meta-learning) and implemented on various NLP tasks. The project that will follow the redaction of this report will aim at evaluating the relevance of some of these approaches to federated learning and NLP research, and exploring the possible improvements that could contribute to the current federated learning scene from an NLP standpoint.

Chapter 2

Theoretical Background

2.1 Federated Learning

In contexts where the training of a central model is hindered by limited access to data due to privacy restrictions on individual devices where the data is stored, decentralised machine learning approaches are used to train models locally and feed into a central model without directly disclosing private data from the device. As Niknam, Dhillon and Reed point out, federated machine learning is an emerging decentralised learning protocol that deals particularly well with privacy and resource constraints [35].

Each model is trained locally on local data and the resulting parameters are sent to the central server, before being updated, taking into account weights from other local models. These new parameters are then fed back to the local models while the whole process preserves the confidentiality of private data. Federated learning has the advantage of reducing the network load thanks to the fact that only features and weights are sent to to server, instead of raw data [6].

Limitations The very nature of federated learning determines some limitations of this method. Given the frequency of communication between nodes required during the learning process, a consistent limitation is that local devices are expected to have a relatively high amount of computing power, local memory and a high bandwidth connection. Another downside of this method is the bias that each node can have towards the whole dataset, given the heterogeneity of local datasets. This heterogeneity is also time-bound in that distribution within the dataset can vary with time. Additionally, biases are difficult to identify given the inaccessibility of the dataset as a whole. Adding to these limitations, vulnerability to some specific attack techniques weakens the data confidentiality property that characterises this method.

Attacks Five attack methods [47] are a particularly threatening to federated learning models specifically:

- *reconstruction attacks*, which attempts to recreate the training dataset by using aggregate statistics about it (e.g. gradient client update or support vector machine);

- *model inversion attacks*, only possible if the model can be queried for a substantial number of times, it would recreate the model through equation solving using the result of such queries;
- *membership-inference attacks*, which consists in building a "shadow model"[1] which is used to predict memberships based on the labeled inputs and outputs of the shadow model itself;
- *attribute inference attacks*, which tries to find the identity of an anonymised user by using other public datasets containing information about such user[2];
- *model poisoning attacks*, which uses unexpected and/or misleading input for purposes of varying severity, from that of lowering the performance of the model up to leveraging the machine learning system and take control over it [29], although the former seems to be more common in federated learning environments [47].

2.1.1 Horizontal vs. vertical

In horizontal federated learning (also known as homogeneous federated learning), datasets share the feature framework, while differing in their samples, while in vertical federated learning (also called heterogeneous federated learning) the ID space is shared across the dataset, but feature spaces differ [63]. As an example, Google GBoard uses horizontal federated learning [64], where the same features are sampled on all devices, while the Chinese WeBank operates an open-source platform supporting vertical federated learning [62].

2.1.2 One-shot federated learning

A technique tackling the limitation of frequent communication requirements of federated learning is introduced by Guha, Talwalkar & Smith [18], who illustrate how to

[1] According to Shokri et. al [47], a shadow model is a model that imitates the behavior of a target model. Its training datasets, unlike that of the target model, is not unknown.

[2] A relatively famous example is that of the Netflix Prize Data breach, which techniques are explained in the technical report "How to Break Anonymity of the Netflix Prize Dataset" [33].

implement one-shot federated learning, a specific type of federated learning allowing a central server to learn a global model over a single round of communication.

Their approach proposes to train a model to completion locally, on each device, and then ensemble the models on the central server by means of one of the following strategies: (1) random selections of k devices, (2) cross-validation selection of devices which achieved a predefined baseline, and (3) selection of models which were trained on some baseline amount of data. This ensemble outputs the final result by averaging the predictions of each model. The resulting central learner is expected to outperform all the local models and, potentially, the full ensemble.

However, the size of the final ensemble might be quite large, so it is necessary to reduce its size in order to transfer it to all the devices for performing inference. The authors propose the distillation-based [20] solution for this problem. It consists in training a new model on some public unlabeled data in order to minimize the difference between its predictions and the ones of a local model. The advantages of this new model are that its size is smaller, it respects the confidentiality of private data and it performs almost as well as the original model even with a small number of examples.

2.1.3 Federated NLP

A promising set of applications of federated learning is represented by natural language processing tasks. When Google first applied the concept of federated learning in the field of NLP (2016), the implementation was developed around the Gboard, a virtual keyboard for touchscreen mobile devices supporting more than 600 language varieties [19].

The importance of efficient input methods is rising with the increase of smartphone usage. Such input methods are nowadays required to include features like auto-correction, word completion, and next-word prediction. Models trained with data gathered from typed text are expected to perform better on these tasks compared to data coming from other sources [2], but accessing the text typed on a device is sometimes problematic due to privacy constraints.

Therefore, federated learning shows a great potential in this field and it contributed to motivate our evaluation of its applicability in the field of NLP, by deploying some of the most effective learning techniques in use to this day (i.e. ensemble learning, meta-learning and transfer-learning) on a federated dataset.

Perhaps because of the increasingly strict privacy legislation and the upsurge in privacy-breaches that involved some well-known multinational corporations, a growing research interest in federated learning, also applied to the field of NLP, resulted in a number of papers being published very recently, yet to be peer-reviewed, such as the ones referenced hereafter in this section.

As an example, an application of federated learning in the field of NLP was implemented for the purpose of keyword spotting. Wake words are used to mark the beginning of an interaction with a voice assistant and are supposed to be detected in a continuous recording of the user's voice. However, being speech considered by nature sensitive data, its centralised collection raises obvious privacy concerns and an attempt to tackle them was made by implementing federated learning [26].

Seen the requirements of modern digital keyboards, another important component of keyboard development drew the attention on the federated learning scene, quite recently: an out-of-vocabulary word learning task was tested on federated datasets reporting to have achieved good recall and precision scores [7].

2.1.4 Federated learning and other learning techniques

In our research we will explore the application of various learning techniques on federated datasets. Some of these techniques have reportedly been successful in their federated form, such as Federated Transfer Learning (FTL) [27, 8], federated meta-learning [14], federated model distillation[**?**], federated average [30] or federated mediation [60].

The Federated Averaging (FedAvg) method consists in training each device locally for several epochs, then sending its weights to the central server, where the weights from all the models are averaged using the following formula:

$$w = \sum \frac{n_k}{n} w_k,$$

where n_k - number of examples on a k^{th} device, n - total number of examples. These averaged weights w are then sent back to each model for the local update.

The Federated Mediation (FedMed) framework uses a combination of FedAvg technique and an Adaptive Aggregation. Adaptive Aggregation consists in defining the importance of each weights not by means of the dataset of a device from which these weights came, but by computing the difference between the local and global weights, as it is assumed that weights which differ the most are the most precious ones. This Adaptive Aggregation technique is combined with FedAvg one using a mediation inceptive scheme, which chooses one of these two strategies to compute the averaged weights, depending on the difference between current and previous loss values.

Meanwhile, no research appears to have been published on the application of some other techniques on federated learning, such as stacking generalisation.

2.2 Ensemble learning

Ensemble learning is a machine learning technique that consists in combining multiple models, henceforth referred to as base-learners, which have been previously trained to solve the same problem. Strategically combining their results aims at improving predictions and obtain an overall better performance compared to that of each of the single models, which makes ensembling a standard approach to improve accuracy in machine learning [11].

Ensemble learning is considered to be based on the idea of "wisdom of the crowd" [44]. This concept can be easily illustrated with an experiment conducted by Francis Galton in the 19th century. The English philosopher and statistician, also known for conceiving the concepts of standard deviation and correlation, is said to have organised a contest during a livestock fair, in which he asked various people to guess the weight of an ox. Although none of the participants guessed the exact weight, the average of the predictions was quite close to the actual value. This experiment proved the power of combining different guesses to achieve a more accurate prediction, an idea subsequently implemented in machine learning through ensemble modelling.

An exhaustive introduction to various methods of ensemble learning was presented in [22], the authors of which name 18 classifier combination schemes. Among the most prominent are different kinds of boosting, averaging, voting, bagging (an alternative to averaging) and stacking.

Boosting is a method that combines several weak classifiers, applying various weights on each of their predictions with the aim of creating a strong classifier. This method was proposed by Schapire [46] as an answer to the question "Can a set of weak learners create a single strong learner?" raised informally in 1984 by Valiant [51] and formally by Kearns in 1988 [24].

In order to obtain a prediction that is better than the one from a single model, bagging [4] trains several classifiers and then calculates the average of their results. Bagging consists in choosing various subsets of a training set using bootstrapping [12] and then training a model on each of these subsets.

Below are more precise descriptions of two ensemble methods that could be used in the implementation phase of our project: a trainable one (stacking) and a not-trainable one (**averaging**).

2.2.1 Averaging

Averaging is one of the simplest techniques in ensemble learning, which consists in training several classifiers on the same dataset and taking the average of their predictions as the final prediction.

When the labels, produced by models are classes, the averaging strategy called voting is used. There are various ways to achieve the agreement of several classifiers. For example, hard voting computes number of predictions for each class and chooses the biggest one, while soft voting computes an average probability of each class and chooses a class with the highest mean probability.

When the labels, produced by models are real values, their average is taken as a final prediction. This method is shown to produce higher result compared to using the prediction of a single classifier [32].

Additionally, there exists a more complex averaging technique that is expressed as a weighted sum of the classifiers, as opposed to a Pythagorean mean.

$$\sum_{j=1}^{p} \alpha_j y_j(\mathbf{x})$$

2.2.2 Stacked generalisation

One of the best performing techniques in ensemble modelling consists in developing an architecture that assembles more than one learning stage, in which base-learners' output is reprocessed by a meta-learner, which learns to make predictions that are more accurate than the ones of the base models.

Known as stacking, this technique is likely to be the most prominent in ensemble learning [59] and its main advantage is that of being able to create a model that is more reliable than any of the single base-learner, to be used successfully in both semi-supervised tasks (with frameworks such as FixMatch [49] or FedMatch [23]) and unsupervised tasks (using, for instance, k-means algorithms [31], or matrix factorization [56]).

2.2.3 Stacking and NLP

As Rajani, Viswanathan et al. suggest, stacking can be successfully applied to NLP tasks. Their *Information Extractors for Knowledge-Base Population* is an example of stacking, combined with other techniques, which achieved state-of-the-art results on a Knowledge Base Population (KBP) English Slot Filling (ESF) task. their ensemble outperformed all other systems in the 2014 KBPESF competition, obtaining an F1 of 48.6%[3] [54].

2.3 Meta-learning

At the beginning of 90s of XX$^{\text{th}}$ century, inspired by the cognitive definition of *meta-learning* - the idea of "being aware of and taking control of one's own learning" [1], computer scientists started applying this concept to machine learning by creating a new technique, also called "meta-learning," aiming at making a model "learn how to learn" without large amounts of data needed. Since the 90s, three main approaches to meta-learning have emerged: optimization-based, metric-based and model-based learning [21].

Optimization-based approach A typical machine learning algorithm is trained by going through a substantial number of optimization steps in order to refine the weights. However, in cases when the data is scarce, an additional meta-model can be implemented to support the learning of the main model. A meta-model is trained to predict all the parameters to be used by the main model as well as the optimization function[4]. This method implies two nested training processes: a meta-model is trained to predict the parameters to be used by the main model; subsequently, these parameters are implemented in the first training step. Then a loss function for the meta-model is computed by means of an error signal deduced from the comparison of the predicted labels and the golden labels for the main task. Once the meta-model's parameters are updated, the next training step is initialised. By repeating the same procedure at every step, the resulting parameters used for training the main model are expected to be optimal and improve the overall quality without the need to train on a large dataset.

Metric-based approach The main idea of a metric-based approach in meta-learning consists in learning a function that correctly measures the similarity between two objects. After building the whole system, it is possible to train it to determine whether two objects belong to the same class or not, on a limited number

[3] The best performing system in the previous competition had reached a maximum F1 of 39.5%

[4] these training processes can be carried out either jointly or independently

of examples, and later use this information for predicting the relationship between examples and classes, even when these were not represented in the training set.

Model-based approach Model-based approach in meta-learning involves using an external memory during the training and predicting processes. A common idea in all model-based approaches is to encode the inputs in some general way and, by comparing them with the already obtained information about the tasks for which an algorithm was trained, choose the best predictor and output the result depending on it. The most popular models in this approach are Neural Turing Machine (NTM) [16] and Memory-Augmented Neural Networks (MANN) [45].

2.3.1 Meta-learning and NLP

MetaNMT, a particularly effective technique based on a popular optimization-based algorithm MAML [15], was proposed in 2018 [17]. The aim of this algorithm is to efficiently solve the task of a neural machine translator on low-resource languages by applying meta-learning techniques. The idea of this system is to train different models on several subsets of both high-resource and low-resource languages in order to obtain the optimal initialization parameters that then could be used for training any model with only a few examples. The main difference between this approach and transfer learning one is the fact that the former takes into consideration how the learning process for low-resource models works too, so the model receives averaged initialization parameters (and not one language biased as it happens in transfer learning). The experiments conducted by the authors showed that this approach outperforms multilingual transfer learning approaches [67].

2.4 Transfer learning

Transfer learning is a frequently used method in machine learning consisting in using a model to complete a different task than the one it was trained for. The concept behind it is based on the assumption that features that were extracted during training for solving a first task may be also useful during the training for another task. The definitions of transfer learning were proposed in 1976 by some Croatian researchers [43] and in 1991 by machine learning specialists from the U.S. [40]. It is said that the quality of the model trained for the second task is the best, with the first and the second tasks being the most similar [65].

2.4.1 Transfer learning and NLP

Transfer learning in NLP is widely represented nowadays with a range of unsupervised models, pretrained on enormous datasets, which are transferred to less-represented tasks. One of the most prominent pretrained models used for transfer learning in NLP is BERT [10], followed by its variations, such as RoBERTa [28], ALBERT [25], StructBERT [57]. When first published, BERT outperformed previous models on eleven NLP tasks (improving, for example, the GLUE Benchmark [55] score by 7.7 percentage points).

Google researchers recently published a paper called *"Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer"* [41], in which they conducted an exhaustive comparison of various transfer learning techniques, training objectives and architectures, introduced a new dataset and a new pretrained model, that unifies all the tasks into a text-to-text generation problem (more details about the model will be given in Chapter 3). They also presented new state of the art results for several NLP benchmarks.

Chapter 3

Evaluation

3.1 Tasks

The following subsections contain a description of the NLP tasks that are most likely to be used to evaluate our methods, together with their benchmarks and datasets. Since the research that will be carried out in the second phase of this project aims at testing different techniques on federated learning, the datasets described in this chapter are all composed of at least 100,000 examples, in order to ease the simulation of data distribution across devices. For this reason, smaller datasets, even if contributing to a state-of-the-art benchmark and/or employed in the mentioned tasks and benchmarks, are not discussed.

3.1.1 Natural Language Inference

Natural language inference (NLI) consists in categorising a sentence representing a hypothesis as true, false, or undetermined, basing this categorisation on whether a "premise" sentence entails the hypothesis, contradicts it or is neutral towards it.

MNLI The Multi-Genre Natural Language Inference Corpus (MNLI) [58] is a dataset used to train models for NLI tasks. Ten different sources, including transcribed speech, government reports, and fiction, were used to assemble the part of the dataset composing the "premise" sentences. This corpus forms a part of the Glue benchmark.

T5-11B model (Text-to-Text Transfer Transformer with 11 billions of parameters) [41] achieves 92.2% of accuracy on MNLI dataset. This model appears to be almost the same transformer as was originally proposed in [52] (its difference consists in simpler layer normalization and different position embedding scheme). This transformer is trained to take as an input a text and to output the prediction in a textual form (for example, input *"mnli premise: I hate pigeons. hypothesis: My feelings towards pigeons are filled with animosity."*, target *"entailment"*). Unsupervised objective that was used for pre-training this transformer is span corruption and the training was performed on the C4 corpus (The Colossal Clean Crawled Corpus was introduced in the same paper).

QNLI The Question-answering Natural Language Inference dataset, being a part of the Glue benchmark, is a dataset created on the basis of the SQuAD collection. While the task of the SQuAD dataset is to determine which part of the question contains an answer to this question, QNLI dataset contains pairs of question texts and their parts and the task is to understand whether the second sentence contains an answer to the question.

State of the art for the QNLI dataset is 99.2 (Accuracy), this performance is shown by the ALBERT model [25]. This model takes the standard BERT architecture with several changes. Firstly, the authors propose to use a factorized embedding parametrization which consists in projection of the one-hot vectors to the lower dimensional embedding space and then passage of this tensor to the hidden space. This heuristics allows to reduce the number of learning parameters. Secondly, the ALBERT model shares feed-forward networks' and attention parameters. Thirdly, instead of training for the next-sentence prediction task, this model is trained for sentence-order prediction task (detecting whether one sentence precedes or follows another one).

SNLI The Stanford Natural Language Inference (SNLI) Corpus [3] is a part of the SentEval benchmark. This dataset contains 570000 sentence pairs each of which is labeled with one of the three following relations: contradiction, entailment or neutral.

Current state-of-the-art accuracy for the SNLI dataset is 92.4%. This result was achieved with the multi-task model [39] that uses BERT and several additional techniques (conditional attention, conditional alignment, conditional layer normalisation and conditional adapters) for the model to show SoTA results on SNLI and other 23 tasks.

3.1.2 Caption-Image Retrieval

Caption-Image retrieval can refer to either *image retrieval*, which consists in ranking images by their relevance for a given query caption, or *caption retrieval*, namely ranking captions by their relevance for a query image.

COCO The COCO dataset, used in the SentEval benchmark, provides a training set of 113k images, aligned with 5 captions each. For each image y paired with a caption x, the relative model aims at learning a compatibility score using a pairwise ranking-loss Lcir(x, y):

$$\sum_{y}\sum_{k}\max(0,\alpha - s(Yy,Ux) + s(Yy,Ux_k))+$$

$$\sum_{x}\sum_{k^1}\max(0,\alpha - s(Ux,Yy) + s(Ux,Yy_{k^1}))$$

State of the art for this dataset is currently achieved with Meshed-Memory Transformer [9] and is equal to 72.8 (BLEU-4). This algorithm uses an encoder-decoder architecture where an encoder is a stacked sequence of encoders each of which contains memory-augmented attention and encoding layer. Memory-augmented attention uses self-attention and a supplementary slot of attention (defined by two learnable matrices) that is able to catch the information about the similarity relations that are found by self-attention. The output of the attention, encapsulated within a residual connection and a layer norm operation, is then applied to a position-wise feed-forward layer (the encoding layer). The output of this layer is also wrapped in a skip connection with a layer norm operation. A decoder is also composed of several decoder blocks each of which contains a meshed cross-attention that takes as an input all the outputs from all the encoder blocks used in the full encoder (A) and a sequence of vectors generated during the previous step (previously generated words) (B) put through the masked self-attention. This meshed attention connects each element of A and B through gated cross attentions. The result of this operator concatenated with B is then modulated using the weighting matrix and sigmoid activation. The outputs of all the applications of this attention on A and B is then summed together. This matrix is then processed by the feed-forward layer (the same as in the encoder part) and both of them are encapsulated within a residual connection and a layer norm operation.

3.1.3 Question Answering

Question Answering is the task of providing an answer to a user's question based on a provided context and abstaining from answering when the context is not sufficient to provide an accurate answer.

ReCORD SuperGlue benchmark contains a dataset, called Reading Comprehension with Commonsense Reasoning Dataset (ReCORD) [66], that consists of more than 120000 passages taken from more than 70000 CNN articles. Each of these passages is accompanied by the query where one of the entities discussed in the passage is masked. The task is to correctly choose the missing entity from the given list.

On the ReCORD dataset, the SoTA result is obtained using LUKE model proposed in [61]. It achieves 91.2% of F1-score. This model is based on transformers (training starts with the pretrained RoBERTa model [28]), and its key differences are the task which is to predict not only tokens, but also entities in a text (unlike all other transformer models) and an entity-aware self-attention mechanism that is able to distinguish tokens from entities.

SQuAD The Stanford Question Answering Dataset [42] is a question-answering dataset from the Glue benchmark populated with paragraphs retrieved from Wikipedia which are assigned to a question each, where the paragraphs contain the answer to the corresponding question. The size of the SQuAD dataset is almost double the one of previously manually labelled reading comprehension datasets and it differs in the fact that its answers are not selected from a list of possible choices, but are derived after considering every possible span of text from the corresponding corpus.

State of the art for this dataset is 96.22% (F1) and it is shown by the T5-11B model [41] which was described above (Section 3.1.1).

Paraphrasing Paraphrasing is a term that can be used to refer to both text generation and text analysis tasks. In the domain of text analysis, it refers to the task of determining whether two strings of text are semantically equivalent, i.e. one the paraphrasis of the other.

QQP The Glue benchmark used the Quora Questions Pairs dataset (QQP), with questions retrieved from the community question-answering website Quora, to carry out the task of confirming or denying whether a pair of questions are semantically equivalent. Both F1 score and accuracy were reported for this task. The dataset is characterised for being unbalanced, consisting of more negative pairs than positive (63% negative). Current state of the art for the QQP dataset is 91% (Accuracy) produced by StructBERT [57]. In spite of being based on BERT's architecture, this model also trains on two additional tasks: predicting the right word order and predicting whether, for a given sentence, another sentence precedes it, follows it or does not occur alongside it (i.e. it was sampled randomly).

Chapter 4

Action Plan

This section provides a preliminary plan of action for the evaluation of the application of *stacking*, *transfer-learning* and *meta-learning* to NLP tasks on a federated dataset.

4.1 Techniques

In this section we are going to describe the approaches that will be tested in our research.

4.1.1 Transfer-learning and federated learning

Hard et al. [19] showed in their work, dedicated to federated learning for GBoard next word prediction, that usage of pre-trained models for parallel training of the models with periodical update of gradients on the central server using parameters from several local devices improves the quality of a model and speeds up the convergence. We are going to try to train one of the popular neural network architectures (such as DNN, LSTM, CNN) on the part of one of our datasets and then use this pre-trained model for federated learning based on techniques such as FedAvg [30] or FedMed [60].

We may also try to fine-tune one of the already existing pre-trained models, such as popular state-of-the-art models like BERT[10], ALBERT[25], StructBERT[57], or RoBERTa[28].

We would recreate two federated learning settings: one where we have several small devices on which we would process computations and a second one where only a few but very large devices would be used for training.

4.1.2 Ensemble learning and federated learning

Stacking Stacking consists in training several models and then using their predictions as an input for one or several meta-learners whose aim is to improve the final quality of a model. For implementing a stacking algorithm in our case, we will firstly train separate models on each device and then send their predictions to the central server. Then a meta-learner will be trained on these predictions and a loss function will be computed using target labels taken from each device (if we suppose that this method will be used in real-life conditions, we could suggest anonymising the devices in order to preserve the privacy of the data). Once all rounds of training have been completed, the final model will be sent to each device for inference. As for device models and a meta-learner we will try some of the following algorithms: dense neural networks, convolutional neural networks or classical machine learning algorithms (Support Vector Machine, Logistic Regression, Random Forest etc.).

Averaging We will also explore how different classical machine learning models perform with various types of averaging applied on their predictions. We will evaluate the mean of all the models trained for the task and only several of them chosen with different strategies. An example of such strategy is proposed in [13] and consists in choosing the best out of 100 trained models and then add to this model one ensemble improving the quality. This selection continues as long as the final quality of the ensemble grows.

4.1.3 Meta-learning and federated learning

Finally some among the following meta-learning techniques will be used on the same federated datasets.

Optimisation-based We would use one of the popular optimization-based meta-learning algorithms like MAML [15], FOMAML [34] or Reptile [34] in order to produce the best initialisation parameters for the final neural network that will be used for solving the task. All of these algorithms consist in training a chosen neural network on different subsets of a training set in order to deduce the best parameters that then will be used for training on the whole dataset. In our case, we will train this algorithm on all our devices and choose the parameters that showed the best result as the final parameters for the neural network.

Metric-based For this approach a Relation Network [50], a Matching Network [53], a Prototypical Network [48] or a Siamese Network [5] can be used to solve a few-shot classification problem on our devices. One of the assumptions we are going to verify is that using a model that has been already trained on one device, will work better on a new device that does not share the name of the classes.

4.2 Tools

All the code, unless otherwise specified, will be written in Python 3. The following open-source Python tools will be used during the implementation phase of the project.

Pandas This library is a user-friendly tool for data analysis and manipulation, presented as a fundamental high-level building block for practical, real world data analysis in Python. [36] We will use this tool to handle all our data throughout the project.

PyTorch This machine learning framework comprises a rich ecosystem of tools and libraries that can be used for building neural networks for solving different NLP tasks and much more. [37] We will use this framework for most of the techniques described in Chapter 2.

Scikit-learn This simple and efficient tool for predictive data analysis is easy to reuse in various contexts. [38] It is built on NumPy, SciPy, and matplotlib and, in our project, it will be used to implement various stacking generalisation algorithms.

4.3 Evaluation

The tasks used to evaluate each technique will be selected from the ones discussed in Chapter 3 together with their associated datasets. In the same chapter, the best performing benchmarks were mentioned, for the results obtained with each technique-task combination to be compared and evaluated.

4.4 Timeline

January At the beginning we will be introduced to the tools used in the implementation phase of our project (e.g. Pytorch, Sklearn, etc.), in order to acquire the knowledge and skills needed to understand and apply the aforementioned techniques.

February-March For two months we will focus on the implementation of new models and on reusing existing models by tailoring them according to our purposes.

April We will then evaluate our models on the test datasets mentioned in Chapter 3 by comparing them to existing benchmarks.

May During this phase we will update the implemented models in order to optimise the results and keep track of the changes we made. We might also experiment with different approaches in case our results were not satisfying.

June Finally, we will record all our results with a detailed description of the implementation in a final report.

Bibliography

[1] J. B. Biggs. The role of metalearning in study processes. *British journal of educational psychology*, 55(3):185–212, 1985.

[2] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecny, S. Mazzocchi, H. B. McMahan, T. Van Overveldt, D. Petrou, D. Ramage, and J. Roselander. Towards federated learning at scale: System design, 2019. cite arxiv:1902.01046.

[3] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[4] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[5] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah. Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688, 1993.

[6] E. by: Peter Kairouz and H. B. McMahan. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1):–, 2021.

[7] M. Chen, R. Mathews, T. Ouyang, and F. Beaufays. Federated learning of out-of-vocabulary words, 2019.

[8] Y. Chen, J. Wang, C. Yu, W. Gao, and X. Qin. Fedhealth: A federated transfer learning framework for wearable healthcare, 2019.

[9] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10578–10587, 2020.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] T. G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[12] B. Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics*, pages 569–593. Springer, 1992.

[13] M. Fajcik, P. Smrz, and L. Burget. BUT-FIT at SemEval-2019 task 7: Determining the rumour stance with pre-trained deep bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1097–1104, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[14] A. Fallah, A. Mokhtari, and A. Ozdaglar. Personalized federated learning: A meta-learning approach, 2020.

[15] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[16] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

[17] J. Gu, Y. Wang, Y. Chen, V. O. K. Li, and K. Cho. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.

[18] N. Guha, A. Talwalkar, and V. Smith. One-shot federated learning. *CoRR*, abs/1902.11175, 2019.

[19] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage. Federated learning for mobile keyboard prediction, 2019.

[20] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[21] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey. Meta-learning in neural networks: A survey, 2020.

[22] A. Jain, R. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22:4–37, 01 2000.

[23] W. Jeong, J. Yoon, E. Yang, and S. J. Hwang. Federated semi-supervised learning with inter-client consistency, 2020.

[24] M. Kearns. Thoughts on hypothesis boosting. *Unpublished manuscript*, 45:105, 1988.

[25] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. Albert: A lite bert for self-supervised learning of language representations. *International Conference on Learning Representations*, 2020.

[26] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau. Federated learning for keyword spotting, 2019.

[27] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang. A secure federated transfer learning framework. *IEEE Intelligent Systems*, 35(4):70–82, 2020.

[28] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[29] Y. Liu, M. Shiqing, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang. Trojaning attack on neural networks. 01 2018.

[30] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[31] P. Mohassel, M. Rosulek, and N. Trieu. Practical privacy-preserving k-means clustering. *Proceedings on Privacy Enhancing Technologies*, 2020(4):414 – 433, 01 Oct. 2020.

[32] U. Naftaly, N. Intrator, and D. Horn. Optimal ensemble averaging of neural networks. *Network: Computation in Neural Systems*, 8(3):283–296, 1997.

[33] A. Narayanan and V. Shmatikov. How To Break Anonymity of the Netflix Prize Dataset. Technical Report cs.CR/0610105, Oct 2006.

[34] A. Nichol, J. Achiam, and J. Schulman. On first-order meta-learning algorithms, 2018.

[35] S. Niknam, H. Dhillon, and J. Reed. Federated learning for wireless communications: Motivation, opportunities, and challenges. *IEEE Communications Magazine*, 58:46–51, 06 2020.

[36] T. pandas development team. pandas-dev/pandas: Pandas, Feb. 2020.

[37] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 8026–8037. Curran Associates, Inc., 2019.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[39] J. Pilault, A. Elhattami, and C. J. Pal. Conditionally adaptive multi-task learning: Improving transfer learning in NLP using fewer parameters & less data. *Computing Research Repository*, abs/2009.09139, 2020.

[40] L. Y. Pratt, J. Mostow, and C. A. Kamm. Direct transfer of learned information among neural networks. In *Proceedings of the Ninth National Conference on Artificial Intelligence - Volume 2*, AAAI'91, page 584–589. AAAI Press, 1991.

[41] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[42] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, Nov. 2016. Association for Computational Linguistics.

[43] A. F. S. Bozinovski. The influence of pattern similarity and transfer of learning upon training of a base perceptron b2. (original in croatian: Utjecaj slicnosti

likova i transfera ucenja na obucavanje baznog perceptrona b2). *Proceedings of Symposium Informatica*, pages 3–121–5, 1976. Bled.

[44] O. Sagi and L. Rokach. Ensemble learning: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(4):e1249, 2018.

[45] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, volume 48, pages 1842–1850, 2016.

[46] R. E. Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

[47] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017.

[48] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017.

[49] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence, 2020.

[50] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[51] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[52] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008, 2017.

[53] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 3637–3645, 2016.

[54] V. Viswanathan, N. F. Rajani, Y. Bentor, and R. Mooney. Stacked ensembles of information extractors for knowledge-base population. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 177–187, Beijing, China, July 2015. Association for Computational Linguistics.

[55] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[56] S. Wang and T.-H. Chang. Federated matrix factorization: Algorithm design and application to data clustering, 2020.

[57] W. Wang, B. Bi, M. Yan, C. Wu, Z. Bao, J. Xia, L. Peng, and L. Si. Structbert: Incorporating language structures into pre-training for deep language understanding. *International Conference on Learning Representations*, 2020.

[58] A. Williams, N. Nangia, and S. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[59] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241 – 259, 1992.

[60] X. Wu, Z. Liang, and J. Wang. Fedmed: A federated learning framework for language modeling. *Sensors*, 20(14):4048, 2020.

[61] I. Yamada, A. Asai, H. Shindo, H. Takeda, and Y. Matsumoto. Luke: Deep contextualized entity representations with entity-aware self-attention. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[62] K. Yang, T. Fan, T. Chen, Y. Shi, and Q. Yang. A quasi-newton method based vertical federated learning framework for logistic regression, 2019.

[63] Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2), Jan. 2019.

[64] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays. Applied federated learning: Improving google keyboard query suggestions, 2018.

[65] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, volume 2, pages 3320–3328, 2014.

[66] S. Zhang, X. Liu, J. Liu, J. Gao, K. Duh, and B. Durme. Record: Bridging the gap between human and machine commonsense reading comprehension, 10 2018.

[67] B. Zoph, D. Yuret, J. May, and K. Knight. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas, Nov. 2016. Association for Computational Linguistics.