Fine-tuning a Swedish BERT with Federated Learning

Dylan Mäenpää (dylma900)
dylma900@student.liu.se
TDDE16 - Text mining

2021-01-13

Abstract

Due to regulatory and private reasons, sharing data between entities can often be hindered. Because of this, existing real-world data is not fully exploited by machine learning (ML). An approach to solve this is to train ML models in a decentralized manner using federated learning (FL). In this project, Swedish BERT models are fine-tuned with FL for the task of multi-class document classification. The data set used are speeches retrieved from the Swedish Parliament. Furthermore, the models are compared to a centralized trained model with respect to precision, accuracy, recall, and macro average F1-score. When the training is split into three locations, the results show that models trained with FL give comparable results to a centralized trained model. Using 4 or 5 locations, a slight performance drop was observed. The decrease in F1-score performance ranged from 0%-6%. It is concluded that FL could be a viable choice for training models with decentralized sensitive data.

1 Introduction

Recent advances in machine learning (ML) have led to highly performing innovations in many fields. For example, in the medical specialty dermatology, ML has been used in skin cancer diagnosing and performed at the same level as dermatologists [3]. Furthermore, the utilization of ML technology could increase efficiency and reduce costs in healthcare [15, 9]. Many recent prominent applications use an ML method called deep learning (DL) [4] that is highly dependent on sufficiently large and diverse data sets to be reliable [11]. However, generating such types of data sets can be difficult. In several domains, data are owned by many entities and stored at different locations. Due to privacy and regulatory reasons, the sharing of data across the entities is hindered, e.g. in healthcare [13]. As a result, it can be difficult to generate robust ML models that have been exposed to diverse data, and consequently, existing data is not fully capitalized by ML.

A solution to train ML models that respect privacy and regulations is by using a decentralized method called federated learning (FL) [7]. In short, using FL, models are trained locally with local data and only the parameters of the updated models are aggregated at a central server. For a more thorough description of FL see section 2.2.

In this project, the aim is to study how the performance of fine-tuned BERT models [2] changes using FL. The models will be fine-tuned for the task of document classification. Recently, fine-tuning of pre-trained DL language models such as BERT has reached strong performance on numerous tasks such as question answering [2]. Following the success of BERT, the National Library of Sweden have released Swedish pre-trained language models [6]. In this project, fine-tuning will be conducted on a Swedish pre-trained BERT. Speeches from the Swedish Parliament will be used, and the goal of the models is to classify parties to speeches.

2 Theory

2.1 Bidirectional Encoder Representations from Transformers

Bidirectional Encoder Representations from Transformers (BERT) [2] is a machine learning method that employs a transformer architecture. The architecture is designed to handle sequential data in parallel which reduces training costs [14]. Moreover, the aim of pre-training BERT is to learn a general-purpose language model. Unlike context-free models such as word2vec [10] that generates a single word-embedding, BERT learns contextual relations between words in a text. Pre-training BERT is expensive, however, fine-tuning a pre-trained model is inexpensive and only a few hours of training on a GPU is needed for achieving a high-performing model on numerous tasks [2]. Also, the modifications to the model for fine-tuning can be simple, e.g. adding a linear layer on top of BERT for handling a classification problem [2].

Compared to previously unidirectional trained models, bidirectional training is employed by BERT resulting in a better understanding of relationships between words [2]. Furthermore, the training process is unsupervised; only a plain text corpus is needed for pre-training. The pre-training process is performed in two ways: 1) some words are masked in the text during training and the model's goal is to predict the masked words based on the text; 2) pairs of sentences are used and the goal of the model is to predict whether the second sentence proceeds the first. Then, the model minimizes the combined loss of the two approaches [2].

Efforts have been made to pre-train Swedish language-models. The Swedish Employment Public Service (in Swedish: Arbetsförmedlingen) and the Swedish National Library (in Swedish: Kungliga biblioteket) have released Swedish pre-trained models based on BERT [6]. The model distributed by the Swedish National Library has been pre-trained on a Swedish text corpus consisting of

1) Digitized newspapers, 2) Official Reports from the Swedish Government, 3) Legal e-deposits, 4) Social media, 5) All Swedish Wikipedia Articles. [6]. Furthermore, the models trained by the Swedish National Library showed superior performance to Googles multilingual models and The Swedish Employment Public Service's models across a range of downstream tasks [6].

2.2 Federated Learning

ML models are normally trained in a centralized manner where the data is stored at one location and the owner of the model can freely observe the data. FL [7] is a technology that decentralizes the generation of an ML model orchestrated by a central server. Using the proposed *FedAvg* algorithm by McMahan et al. [7] the training of a global model is done in cycles, where a cycle starts with the central server distributing a global model to participating locations. Then each location trains the identical models with local data and sends updated local model parameters to the central server. Hence the raw local data is not observable by the central server. Further, a global model is updated by aggregating and averaging the local model parameter updates, effectively ending a cycle. Formally the global parameter update is given by:

$$P_g^c = \sum_{k=1}^{K} \frac{n_k}{N} P_k^c \tag{1}$$

Where P_g^c is the global model parameter values at cycle c, K is the number of locations, n_k is the number of samples at location k, N is the total number of samples at all locations and P_k^c is the local model parameters at k at cycle c.

3 Data

Freely available speeches from the Swedish Parliament[1] from the year 2017/2018 and 2018/2019 were used. The speeches were given by eight different parties: Centre Party (C); Christian Democrats (KD); Liberals (L); Social Democrats (S); Sweden Democrats (SD); Green Party (MP); Moderate Party (M); and Left party (V). Moreover, the speeches were first tokenized using a tokenizer given by the course. The tokenizer employed regular expressions and turned the tokens into lowercase characters. Then, following the input requirements to BERT, all speeches were further tokenized using the tokenizer from the Swedish BERT models authors [6]. The total amount of speeches were 21 631, and the data were split up into a training set and test set. 80% of the total speeches were randomly chosen for the training set and 20% for the test set. Furthermore, the distribution of the total speeches over the parties and statistical analysis of the number of tokens in the speeches can be seen in figure 1.

[1]Data available at: https://data.riksdagen.se/in-english/

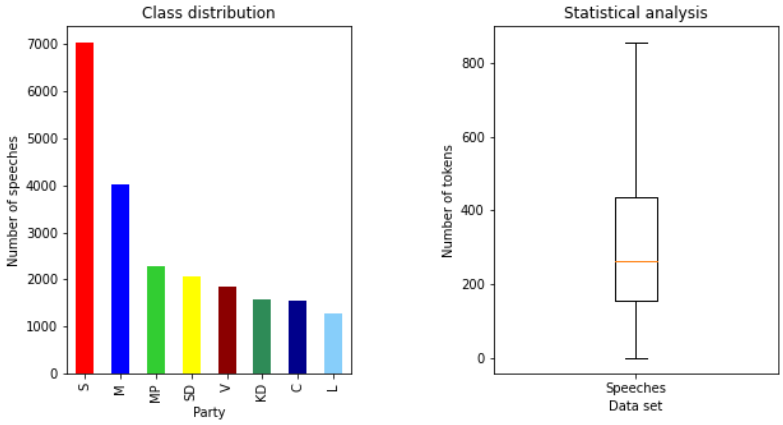


Figure 1: To the left, the distribution of the speeches over the parties in the data. To the right, a statistical analysis of the number of tokens in the data. The orange line represents the median number of tokens. Boxes cover the 25th percentile and 75th percentile. The whiskers cover the 10th and 90th percentile.

4 Method

To study the performance of a model trained with FL, an equivalent baseline centralized model was trained, see section 4.1. The centralized model is then compared to models trained with FL, see 4.2. A total of 5 experiments were conducted. Note that the same total amount of data were used in all experiments. Furthermore, the code used for conducting the experiments can be found at Github[2].

4.1 Centralized trained model

Following Devlin et al. [2] in adapting BERT for document classification; a dropout layer and a fully connected linear layer were added on top of a pre-trained Swedish BERT [6], namely *bert-base-swedish-cased*[3]. The linear layer takes 768 features as input, which conforms to the pooled output of the BERT, and outputs 8 features (the number of parties that have given speeches). Following [12], hyperparameters were chosen. Cross entropy loss is adopted. The dropout probability was set to 0.1. Adam was employed as optimizer with parameters $B_1 = 0.9$, $B_1 = 0.999$, learning rate of 5e-5, warm-up proportion of 0.1 and a L2 weight decay of 0.01. Furthermore, due to scarce computational resources, only the last 254 tokens of a speech were used. Further, as required by BERT when fine-tuning the model, two tokens were added for every sample, [CLS] is prepended and *[SEP]* appended in every speech. If a speech was less

[2]https://github.com/dylanmaenpaa/tdde16-project
[3]Model can be found at: https://github.com/Kungbib/swedish-bert-models

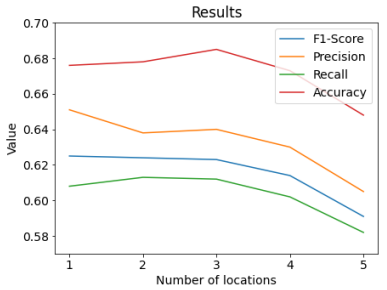
than 210 words, it was padded with *[PAD]* tokens. The training batch size was 8 and the number of epochs was set to 10. Moreover, the model was trained end to end. At the end of each epoch, the precision, recall, accuracy, and macro average F1-score were calculated on the test set. All computation was done using the free GPUs provided by Google Colab[4]. The duration of the training was approximately 3 hours.

4.2 Models trained with FL

The FL process was simulated on Google Colab and trained with the free GPUs. To mimic N different locations, the training data were randomly split into N equally large sets. Then, following the FL algorithm *FedAvg*, N identical models were trained with local data for 10 cycles. In a cycle, each location trained the distributed model for one epoch with the local data. After each cycle, the precision, recall, accuracy, and macro average F1-score of the test set were calculated with the global updated model. A total of 4 global models were trained where N was set to 2, 3, 4, and 5. The same model settings described in 4.1 were employed for all models. Furthermore, for each experiment, the training time was approximately 3 hours.

5 Results

The results from the 5 conducted experiments can be seen in figure 2. It can be observed that the decrease in performance in macro average F1-score ranged from 0%-6%.



Number of locations

Metric	1	2	3	4.	5
F1-score	0.625	0.624	0.623	0.614	0.591
Precision	0.651	0.638	0.64	0.63	0.605
Recall	0.608	0.613	0.612	0.602	0.582
Accuracy	0.676	0.678	0.685	0.673	0.648

Figure 2: Experimental results after 10 epochs/cycles for 1-5 simulated locations. One location represents the centralized model, locations 2-5 represents the models trained with FL.

[4]https://colab.research.google.com/

6 Discussion

The results show that when the data set is distributed to up to three locations the performance of the trained models with respect to macro average F1-score is almost identical. This suggests that the performance of a fine-tuned BERT trained with FL in document classification could achieve at least the same performance as a centralized trained model. However, as the number of locations increases the results show a slight downward trend of all evaluation metrics.

Results from this project are similar to the results from related work by Liu et al. [5] where a performance drop was observed when fine-tuning BERT using FL with 5 locations. The F1-score performance drop ranged from 2%-6% depending on how they conducted the pre-training of BERT. Compared to this project, there were several differences in the study conducted by Liu et al [5], and the three most major differences follow. Firstly, the model and training settings differed. Secondly, their BERT was fine-tuned to perform the task of Named Entity Recognition. Thirdly, they used clinical notes for fine-tuning, i.e. the data set was dissimilar. Despite these differences, similar results in this project were obtained. Furthermore, in the study conducted by Mcmahan et al. [7], the experiments showed superior performance in models trained with FL compared to a centralized trained model, arguing that the FL process could act as a regularization method similar to dropout. Thus we reason that the performance of fine-tuned BERT models trained with FL depends on many factors, such as what pre-trained model is used, and what data set is used. Also, we find that our results and the results from the aforementioned related work [5, 7] suggest that FL can be a viable method for fine-tuning BERT models with decentralized data.

Often in real-world cases, there is data heterogeneity among locations. For example in a healthcare environment, two organizations might collect data from an older respective younger population. This data distribution disparity could impact the performance of an ML model trained with FL. This is discussed by Mcmahan et al. [7], and they empirically show that a convolutional neural network (CNN) trained with heterogeneous data over locations achieves results comparable to a CNN trained with non-heterogenous data. The objective of the CNN models was image-classification (classifying digits). As future work, it would be interesting to see how well a BERT performs when fine-tuned with FL and with heterogeneous data over locations. For example, a location could hold speeches only from The Social Democrats, and another location could holds speeches only from the Moderate Party.

There are several limitations to this project. Extensive hyper-parameter tuning has not been conducted. Also, due to scarce computational resources, only the last 254 words of the speeches were used for training. As seen in figure 1, more than 50% of the data contains more than approximately 270 words, hence some words are not used in training for many speeches. Furthermore, it has been shown that private information can be exposed in the trained models [1]. By utilizing privacy measures such as differential privacy [8] or secure multiparty communication (MPC) [1], privacy can be preserved with the cost of

computational efficiency or performance. Thus, other results might have been acquired if differential privacy or MPC were used in conjunction with FL in this project.

7 Conclusion

In this project, fine-tuning of Swedish BERTs with FL have been conducted in a centralized and decentralized manner. The centralized trained model acted as a baseline. Using a decentralized approach, models were trained using FL on 2-5 locations. The objective of the models was to classify speeches to political parties. An evaluation of the models was conducted with respect to performance (accuracy, precision, recall, and macro average F1-score). The macro average F1-score performance decrease ranged from 0%-6% compared to the baseline. The results showed that the performance of the models trained with FL with up to three locations was comparable to the baseline. Then a slight downward trend was observed in the results. Furthermore, this project suggests that fine-tuning BERT models with FL could achieve viable performance comparable to a centralized trained model. Hence it is concluded that FL is a promising approach in fine-tuning BERT, and could be a fitting method in training ML models when dealing with private and regulatory data sets concerns.

References

[1] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

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

[3] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

[4] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[5] Dianbo Liu and Tim Miller. Federated pretraining and fine tuning of bert using clinical notes from multiple silos. *arXiv preprint arXiv:2002.08562*, 2020.

[6] Martin Malmsten, Love Börjeson, and Chris Haffenden. Playing with words at the national library of sweden – making a swedish bert, 2020.

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

[8] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

[9] Bertalan Meskó, Gergely Hetényi, and Zsuzsanna Győrffy. Will artificial intelligence solve the human resource crisis in healthcare? *BMC health services research*, 18(1):545, 2018.

[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[11] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[12] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer, 2019.

[13] Willem G Van Panhuis, Proma Paul, Claudia Emerson, John Grefenstette, Richard Wilder, Abraham J Herbst, David Heymann, and Donald S Burke. A systematic review of barriers to data sharing in public health. *BMC public health*, 14(1):1–9, 2014.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[15] Kun-Hsing Yu, Andrew L Beam, and Isaac S Kohane. Artificial intelligence in healthcare. *Nature biomedical engineering*, 2(10):719–731, 2018.