Toward Federated Learning Through Intent Detection Research

Daiga Deksne[1,2][0000−0002−8916−0320], Jurgita Kapočiūtė-Dzikienė[3,4][0000−0002−8402−4549], and Raivis Skadiņš[1,2][0000−0003−0929−2380]

[1] Tilde, Vienības str. 75A, Riga, Latvia
[2] Faculty of Computing, University of Latvia, Raiņa bulv. 19, Riga, Latvia
[3] Tilde IT, J. Jasinskio str. 12, Vilnius, Lithuania
[4] Faculty of Informatics, Vytautas Magnus University, Universiteto str. 10, Kaunas, Lithuania
{daiga.deksne,raivis.skadins}@tilde.com, jurgita.dzikiene@tilde.lt

Abstract. Modern organizational communication heavily relies on virtual assistants, necessitating robust Natural Language Understanding (NLU) models for effective interaction. This research tackles the challenges of NLU model development across multiple languages, including Estonian, English, German, Spanish, French, Italian, and Latvian. We explore various intent detection methodologies, including memory-based techniques that encompass both vectorization with LaBSE, ADA, or SONAR models, and semantic search using similarity or Levenshtein distance-based approaches. We explore various intent detection methodologies, including memory-based techniques that encompass both vectorization with LaBSE, ADA, or SONAR models, and semantic search using similarity or Levenshtein distance-based approaches. Additionally, we investigate supervised text classification methods such as FastText+CNN, LaBSE + FFNN, or fine-tuning LaBSE, as well as text generation techniques leveraging OpenAI's Davinci large language model. Our findings highlight the efficacy of memory-based approaches, particularly for non-English languages. We showcase the effectiveness of multilingual and cross-lingual LaBSE vectorization and the SONAR large language model. Furthermore, we introduce open-source intent detection software tailored for Federated Learning (FL). Through a prototype, we demonstrate the seamless integration of this framework into RASA-based virtual assistants, offering practical guidance for organizations interested in deploying intelligent and privacy-preserving conversational agents. This research advances virtual assistant development and highlights the potential of FL for seamless integration with NLU models. In the future, we are planning to test it with various languages and with real client scenarios.

Keywords: Intent detection · Memory-based, supervised classification and generation approaches · Estonian, English, German, Spanish, French, Italian, and Latvian languages · Federated learning.

1 Introduction

In today's digital era, virtual assistants are vital for seamless communication, offering support, and enhancing efficiency.

However, the development of reliable and adaptable virtual assistants poses challenges, especially in crafting Natural Language Understanding (NLU) models. Accurately grasping user intents and delivering appropriate responses necessitates sophisticated NLU models adept at interpreting diverse linguistic inputs.

This project aims to address the complexities of NLU model development for virtual assistants and explore FL for data privacy. Our goal is twofold: 1) to train accurate decentralized bots for individual organizations while keeping their private data decentralized in the original locations, and 2) to create an aggregated centralized bot that serves the collective needs of all participating entities. By combining embedding models, intent detection methods, and FL frameworks, we aim to enhance virtual assistant development outcomes. This paper presents the methodology, experiments, results, and conclusions, highlighting the potential of FL in NLU model development and its implications for the future of virtual assistants. Additionally, we introduce open-source intent detection software based on FL, designed to preserve data privacy. We have also developed a prototype demonstrating the integration of this framework into RASA-based virtual assistants, including popular platforms like Bürokratt.[5] This prototype validates our approach and offers practical guidance for integrating FL into existing virtual assistant systems, enabling organizations to deploy more intelligent and privacy-preserving conversational chatbots.

2 Related work

FL is gaining prominence in Natural Language Processing (NLP), providing a flexible method for training models across various applications. Renowned for boosting data privacy by enabling decentralized model training without centralizing sensitive data, FL's benefits extend further. It plays a crucial role in developing advanced NLP models, enhancing their ability to comprehend and generate human language with remarkable precision and sophistication.

A key application of FL in NLP is training large language models (LLMs)[10],[5], [6], leveraging extensive datasets for tasks such as text classification, translation, and generation. FL aggregates data from varied sources to enhance these models, ensuring privacy. Besides training LLMs, FL supports a wide range of NLP tasks, from text classification to question-answering, enabling model adaptation to downstream tasks [23], [24]. It's particularly crucial for applications requiring stringent data privacy, like virtual assistant development[8].

The FL ecosystem is abundant with NLP-focused frameworks and libraries, each tackling unique challenges in decentralized data processing. TensorFlow

[5] More about this project is on: https://commission.europa.eu/projects/burokratt-programme-and-national-virtual-assistant-platform-and-ecosystem_en.

Federated (TFF) by Google [1] supports NLP tasks while ensuring data privacy on local devices. PySyft [26] enhances PyTorch and TensorFlow with secure computation for NLP privacy. FATE [19] offers a secure, collaborative NLP model training framework. Flower [2] and FLUTE [12] prioritize flexibility, performance, and rapid prototyping with features for optimization and scalability. FedNLP [18] benchmarks FL in NLP, integrating transformer models across varied data settings, while FedAdapter [4] enhances FedNLP's training efficiency with innovative strategies. However, their focus on major languages like English points to a gap in linguistic diversity.

In the dynamic field of FL, specialized algorithms like FedMA [25], FedAvg [20], FedAvgM, FedAdam [22], FedOPT [21], FedProx [17], and AdaPL [7] play crucial roles in addressing the complexities of training across distributed networks. Designed mainly for neural networks, these algorithms leverage gradient descent and optimization to tackle non-IID data challenges and adapt learning rates. FedMA excels in layer-by-layer global model construction, optimizing for varied data distributions with an adaptive weighting system. FedAvg, the cornerstone of parameter aggregation, is enhanced by FedAvgM's meta-learning for tailored training. FedAdam and FedOPT adjust learning rates and momentum for better NLP task performance, while FedProx enhances FL stability in non-convex settings with a proximal term. AdaFL further refines FL by optimizing client selection and participation, aiming at communication efficiency and stable model performance.

Yet, the exploration of FL extends beyond these gradient-based methods. There is an emerging interest in applying FL to memory-based models, a domain ripe for innovation but still largely unexplored. Memory-based models, essential for tasks requiring the quick recall of specific instances or attributes, pose distinct challenges in an FL context. Our research attempts to meld FL's decentralized, privacy-centric advantages with the detailed, instance-specific strengths of memory-based approaches. Furthermore, we seek to broaden the scope of FL into the domain of intent detection, with a focus on enhancing language models for underrepresented languages, such as Estonian.

3 Solution Architecture

This project aims to develop tailored aggregated bots for multiple organizations, combining independent bot creation to construct a unified bot for a seamless user experience. This unified approach addresses diverse domain and organization-related questions, ensuring users don't distinguish between different bots.

Our solution architecture (Figure 1) includes remote bot training environments where trainers develop bots independently. These trainers manage private training data and tailor local models to their bot's specific needs.

Additionally, a central training hub employs a federated approach to train an aggregated NLU model, amalgamating parameters from remote locations. This creates a unified NLU model proficient in discerning intents from shared training data, meeting various organizational requirements.

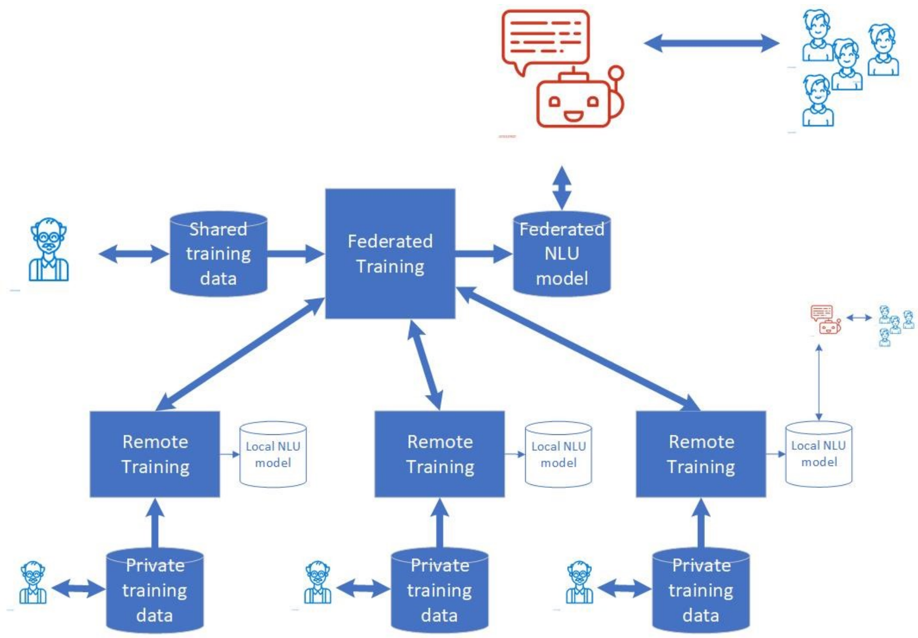

Fig. 1. System overview

4 Intent Detection

The essence of any virtual assistant lies in its NLU module, which can operate either in intent detection or answer generation mode. However, intent detection typically achieves higher accuracy levels with less training data, making it a better fit for our customer expectations.

Consequently, our problem-solving strategy prioritizes intent detection, particularly through supervised text classification, which necessitates the availability of a training dataset.

4.1 Dataset

Our intent detection research aimed to test the effectiveness of various text classification approaches across multiple languages by exploring the potential for integrating additional languages into the NLU model in the future. These experiments were conducted using two datasets:

– *Multi-language segregated dataset* [6] (for the statistics see Table 1) containing several languages (English, German, French, Italian, Spanish, and Latvian).

[6] The dataset has been created in the StairwAI project (`https://stairwai.nws.cs.unibo.it/`) funded by the European Union's Horizon 2020 research and innovation programme under grant agreement 101017142.

The dataset contains 37 intents (classes) and is structured to ensure that each text instance is associated with only one class, assuring the nature of single-label classification. All instances were shuffled and split into training (80%) and testing (20%) subsets within the frame of each class and for each language. The dataset is well-balanced resulting in low majority (1) and random baselines (2).

– *Estonian datasets* containing purely Estonian texts. These three datasets (for the statistics see Table 2) [7] were constructed using data provided by RIA from the Bürokratt project. Instances in each dataset were divided using the same methodology as the *Multi-language segregated dataset*. However, for training and testing, we employed cross-validation. The Sotsiaalkindlustusamet dataset exhibits less balance, leading to a notably high majority baseline that our methods must surpass.

$$majority_{baseline} = max(P_i) \tag{1}$$

, where P_i is the probability of the class.

$$random_{baseline} = \sum (P_i)^2 \tag{2}$$

Table 1. Statistics about the *Multi-language segregated dataset*.

Language	**Instances in training split**	**Instances in testing split**	**Majority baseline**	**Random baseline**
English	386	94	0.074	0.033
German	192	47	0.064	0.032
Spanish	193	47	0.064	0.032
French	193	47	0.064	0.032
Italian	193	47	0.064	0.032
Latvian	183	46	0.065	0.031

4.2 Approaches

Within the frame of this project, we have tested the following text classification methods:

– **FastText+CNN**.This supervised text classification approach combines FastText embeddings [14] with a customized CNN architecture [16]. FastText utilizes subword embedding information, enabling it to construct word vectors even in cases of out-of-vocabulary words or typos. On the other hand,

[7] All Estonian datasets have been made publicly accessible on `https://github.com/tilde-nlp/fnlu/tree/main/Other` assuring transparency and their usage by others in the future.

Table 2. Statistics of the *Estonian datasets.*

Dataset	Number of intents	Number of instances	Majority baseline	Random baseline
Rahvusraamatukogu (National Library)	36	1 104	0.130	0.053
Sotsiaalkindlustusamet (Social Insurance)	7	79	0.418	0.235
Kriisijuhtimine (Crisis Management)	23	287	0.105	0.054

CNN focuses on identifying token n-grams, which is particularly useful for text classification tasks reliant on keywords rather than the entire contextual meaning of the text. For our experiments, we utilized the proprietary FastText embedding model supporting English and Latvian languages in conjunction with the CNN architecture presented in [15]. This method will exclusively serve as the baseline.

- **LaBSE+FFNN**. We utilized the frozen Language Agnostic BERT Sentence Embedding (LaBSE) approach proposed by [11], alongside a two-layer fully connected Feed Forward Neural Network (FFNN) model optimized and fine-tuned for our intent detection tasks. LaBSE differs from traditional BERT embeddings by providing sentence-level representations, capturing the semantics of entire texts simultaneously. It supports 109 languages, including all our target languages, generating fixed-size vectors for texts without retaining word boundaries. LaBSE is not sensitive to word order, therefore it is well-suited for languages with flexible sentence structures, requiring less training data to cover various sentence structures. However, LaBSE's cross-lingual capability may vary across languages.
- **LaBSE-LangChain-k1** (which is the memory-based approach) leverages the LangChain framework, enabling context-aware applications without the need for training. Once training instances are vectorized and stored in the Chroma vector database, the method employs cosine similarity and a greedy search to find the closest training instance to the testing one, thereby assigning its label. LaBSE serves as the vectorization model.
- **LaBSE-LangChain-k10-mv**. This method closely resembles LaBSE-LangChain-k1, but instead of searching for a single similar instance, it searches for the 10 closest instances, collects their class labels, and conducts a majority vote to determine the final class.
- **ADA-LangChain-k1**. This approach resembles LaBSE-LangChain-k1, but instead of LaBSE, OpenAI's text-embedding-ada-002 second generation model [13] is used to vectorize texts instead.
- **ADA-LangChain-k10-mv**. This approach is similar to LaBSE-LangChain-k10-mv, but instead of LaBSE, it uses text-embedding-ada-002 embeddings.
- **Davinci-fine-tuning**. For this approach, we utilized OpenAI's davinci-002 large language model [3], a generative transformer model. In our experiments,

we fine-tuned the added layers' parameters while keeping the hyperparameter values at their defaults. Additionally, we restricted this model to generate only a single token as the input text's label.

– **SONAR-LangChain-k1** adopts a similar approach to LaBSE-LangChain-k1 but: 1) employs the SONAR vectorization model [9] developed by META research instead of LaBSE; 2) uses the Faiss database instead of Chroma; 3) in case of the semantic search, applies the Euclidean distance measure metric instead of cosine similarity. SONAR is an open-source large language model that supports all our target languages and was chosen to broaden the scope of our experimentation, as it has recently garnered significant attention in numerous text classification tasks across various languages. Faiss, on the other hand, is optimal in scenarios that require the combination of multiple indexes into a single retriever, which can contribute to the easier implementation of the FL paradigm.

4.3 Experiments and Results

The initial experiments utilized a *Multi-language segregated dataset*. For methods involving training, the dataset was shuffled and divided into training (80%) and validation (20%) sets, with accuracy chosen as the primary evaluation metric.

Approaches with randomized parameter initialization were tested five times and averaged results along with confidence intervals were calculated. However, this procedure did not apply to methods tested for comparison purposes only, such as FastText+CNN, which serves as a baseline supporting only two languages, or Davinci-fine-tuning, a method that stores models on third-party servers and incurs charges, both of which are unacceptable for our clients. Results obtained with the *Multi-language segregated dataset* are summarized in Fig. 2.

Zooming in on Fig. 2, we can draw the following conclusions. Davinci-fine-tuning, followed by LaBSE-LangChain-k1 and ADA-LangChain-k1, emerged as the most suitable approaches for the English dataset. In contrast, LaBSE-fine-tuning exhibited the highest accuracy for other languages (German, Spanish, French, Italian, and Latvian). Despite its superior performance, LaBSE-fine-tuning proves impractical due to its lengthy training times (up to 1 hour on our small dataset) and high hardware demands (exceeding 12 GiB of GPU RAM). Considering the distributed nature of NLU models training across different locations, it's evident that not all participating entities will possess these computational capabilities. Consequently, LaBSE-LangChain-k1 emerges as the next best option, delivering good results for the majority of non-English languages.

For the second set of experiments with the *Estonian datasets*, we narrowed down the list of tested approaches to LaBSE-LangChain-k1 and SONAR-LangChain-k1. Additionally, we conducted a more comprehensive investigation, evaluating not only accuracy but also precision, recall, MicroF1, and MacroF1. The results in Table 3 demonstrate the slight superiority of SONAR-LangChain-k1 over LaBSE-LangChain-k1. Both LaBSE and SONAR have their advantages and drawbacks in terms of speed and resource usage. LaBSE is faster, completing requests in 27-30 milliseconds compared to SONAR's 120-130 milliseconds,

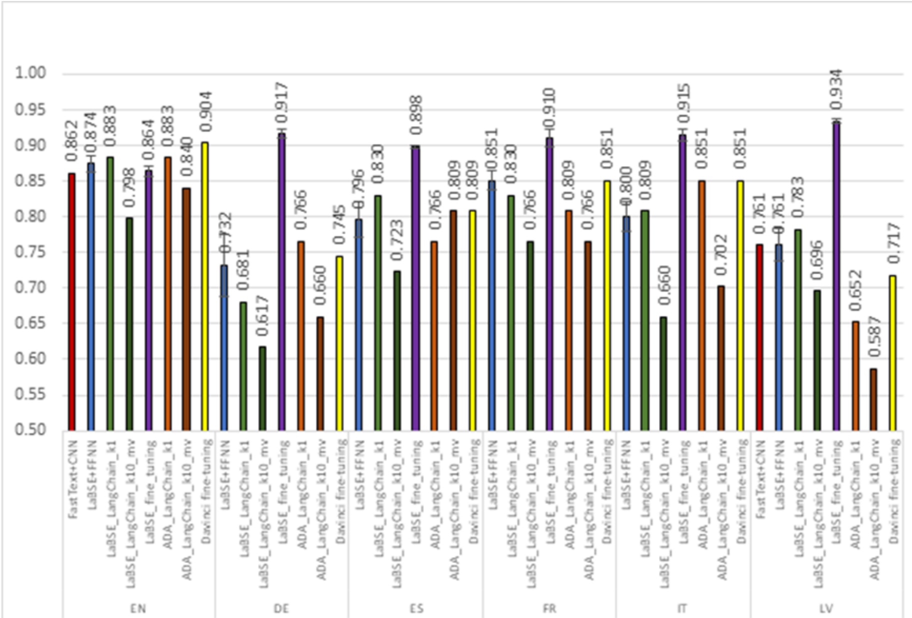

Fig. 2. The accuracy values achieved with different approaches and languages on the *Multi-language segregated dataset* with 37 intents.

but SONAR consumes less disk space at 2.85 GiB, while LaBSE requires 5.27 GiB.

4.4 Discussion

The experiments with a *Multi-language segregated dataset* demonstrate that all tested approaches are suitable for our problem-solving needs, as the achieved accuracies significantly outperform random and majority baselines. Davinci-fine-tuning emerges as the most accurate technique for the well-supported English language, while LaBSE-fine-tuning (with unfrozen parameters and additional layers) proves to be the best technique for all other tested non-English languages (German, Spanish, French, Italian, and Latvian). This outcome is unsurprising, considering that this model is not only multilingual (supporting all our target languages) but also cross-lingual (capable of benefiting from the inclusion of other languages in the model). Despite its significantly good performance, LaBSE-fine-tuning is less practical compared to the second-best approach, LaBSE-LangChain-k1, which requires much less training time and GPU computational resources, making it better suited to the needs of the participating entities in the context of FL.

The experiments conducted with the *Estonian datasets* reveal a slight superiority of the SONAR-LangChain-k1 approach over LaBSE-LangChain-k1. This

Table 3. Evaluation results on *Estonian datasets*.

Dataset	**LaBSE-LangChain-k1**	**SONAR-LangChain-k1**
Rahvusraamatukogu (National Library) 36 intents	accuracy: 0.760 precision: 0.760 recall: 0.760 MicroF1: 0.760 MacroF1: 0.726	accuracy: 0.763 precision: 0.763 recall: 0.763 MicroF1: 0.763 MacroF1: 0.758
Sotsiaalkindlustusamet (Social Insurance) 7 intents	accuracy: 0.608 precision: 0.608 recall: 0608 MicroF1: 0.608 MacroF1: 0.589	accuracy: 0.709 precision: 0.709 recall: 0.709 MicroF1: 0.709 MacroF1: 0.638
Kriisijuhtimine (Crisis management) 23 intents	accuracy: 0.5 precision: 0.5 recall: 0.5 MicroF1: 0.5 MacroF1: 0.484	accuracy: 0.503 precision: 0.503 recall: 0.503 MicroF1: 0.503 MacroF1: 0.479

outcome isn't surprising, considering that the SONAR model has demonstrated strong performance across various text classification tasks in different languages. Our research confirms its effectiveness also for the Estonian language. However, it's worth noting that all datasets used in our experiments are rather small, which complicates the evaluation of statistically significant differences in results. Nonetheless, despite the small size of the datasets, they align with our customers' expectations, and our research demonstrates that favorable outcomes are achievable even with limited data.

Given that memory-based approaches have demonstrated their effectiveness in accurately addressing our intent detection problems, they are naturally well-suited for integration within the FL framework.

5 Solution Implementation

The intent detection experiments underscored the effectiveness of memory-based approaches, particularly LaBSE-LangChain-k1 and SONAR-LangChain-k1 methods, with the latter showing slight superiority on *Estonian datasets*. In memory-based approaches, training entails storing instances and conducting semantic searches, offering several advantages. It eliminates the need for predefined intents during centralized model training and allows the model to adapt to fluctuations in training data. Furthermore, errors in the data have minimal impact on the model's integrity, as intents are independent across participating entities. Each entity can introduce or modify intents, re-vectorize local models, and update the centralized model autonomously, mitigating common issues in standard FL algorithms.

Our federated intent detection solution with the core of the memory-based NLU model is implemented as open-source software, accessible on GitHub[8] under the Apache 2.0 license. It includes two components: a federated intent detector used for both training and real-time inference, and vectorization services offered through Docker containers with options for LaBSE and SONAR embedding models. The solution has undergone the initial testing with the *Estonian datasets*.

The prototype showcases an example implementation integrating FL into Rasa[9] bot software, with setup instructions for configuring and deploying the federated NLU system in distributed environments. This approach can be adapted for integration into other Rasa-based software products like Bürokratt.

Additionally, there are also setup instructions detailing how to set up the federated NLU system with one central site and multiple remote sites across participating entities. These instructions serve as a comprehensive guide for configuring and deploying the system in a distributed environment.

6 Conclusions

Our project aims to develop customized virtual assistants for diverse organizational needs, utilizing innovative NLU solutions. We explored various intent detection methodologies (memory-based, supervised classification, text generation), highlighting the effectiveness of memory-based techniques (such as vectorization of text and semantical search over them), particularly for non-English languages like Estonian. This aligns with our focus and suggests the potential for seamless implementation in FL frameworks.

Additionally, we introduce open-source intent detection software employing FL, enhancing accuracy while prioritizing data privacy and security. A prototype showcases its integration into RASA-based virtual assistants, including platforms like Bürokratt.

In the future, we plan to conduct extensive testing of the FL system in real-world environments, encompassing diverse languages and client scenarios.

Acknowledgments. This research has been supported by "Eesti keeletehnoloogia 2018–2027" project: EKTB78 Liitõppe rakendamise võimalused dialoogiandmete põhjal.

References

1. Authors, T.T.F.: TensorFlow Federated (Dec 2018), `https://github.com/tensorflow/federated`
2. Beutel, D.J., Topal, T., Mathur, A., Qiu, X., Parcollet, T., Lane, N.D.: Flower: A friendly federated learning research framework. CoRR **abs/2007.14390** (2020), `https://arxiv.org/abs/2007.14390`

[8] https://github.com/tilde-nlp/fnlu

[9] https://www.kratid.ee/en/burokratt

3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
4. Cai, D., Wu, Y., Wang, S., Lin, F.X., Xu, M.: Fedadapter: Efficient federated learning for modern nlp. In: Proceedings of the 29th Annual International Conference on Mobile Computing and Networking. ACM MobiCom '23, Association for Computing Machinery, New York, NY, USA (2023). `https://doi.org/10.1145/3570361.3592505`, `https://doi.org/10.1145/3570361.3592505`
5. Che, T., Liu, J., Zhou, Y., Ren, J., Zhou, J., Sheng, V., Dai, H., Dou, D.: Federated learning of large language models with parameter-efficient prompt tuning and adaptive optimization. In: Bouamor, H., Pino, J., Bali, K. (eds.) Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 7871–7888. Association for Computational Linguistics, Singapore (Dec 2023). `https://doi.org/10.18653/v1/2023.emnlp-main.488`, `https://aclanthology.org/2023.emnlp-main.488`
6. Chen, C., Feng, X., Zhou, J., Yin, J., Zheng, X.: Federated large language model: A position paper (2023)
7. Chen, Z., Chong, K.F.E., Quek, T.Q.S.: Dynamic attention-based communication-efficient federated learning. CoRR **abs/2108.05765** (2021), `https://arxiv.org/abs/2108.05765`
8. Cureton, M., Corradini, A.: Federated learning for intent classification. In: 2023 IEEE 19th International Conference on Intelligent Computer Communication and Processing (ICCP). pp. 315–322 (2023). `https://doi.org/10.1109/ICCP60212.2023.10398713`
9. Duquenne, P.A., Schwenk, H., Sagot, B.: Sonar: Sentence-level multimodal and language-agnostic representations (2023)
10. Fan, T., Kang, Y., Ma, G., Chen, W., Wei, W., Fan, L., Yang, Q.: Fate-llm: A industrial grade federated learning framework for large language models (2023)
11. Feng, F., Yang, Y., Cer, D., Arivazhagan, N., Wang, W.: Language-agnostic BERT sentence embedding. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 878–891. Association for Computational Linguistics, Dublin, Ireland (May 2022). `https://doi.org/10.18653/v1/2022.acl-long.62`, `https://aclanthology.org/2022.acl-long.62`
12. Garcia, M.H., Manoel, A., Diaz, D.M., Mireshghallah, F., Sim, R., Dimitriadis, D.: Flute: A scalable, extensible framework for high-performance federated learning simulations (2022)
13. Greene, R., Sanders, T., Weng, L., Neelakantan, A.: New and improved embedding model. Available at `https://openai.com/blog/new-and-improved-embedding-model`, last accessed 2024/02/13 (December 2022)
14. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Lapata, M., Blunsom, P., Koller, A. (eds.) Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. pp. 427–431. Association for Computational Linguistics, Valencia, Spain (Apr 2017), `https://aclanthology.org/E17-2068`
15. Kapočiūtė-Dzikienė, J., Balodis, K., Skadiņš, R.: Intent detection problem solving via automatic dnn hyperparameter optimization. Applied Sciences **10**(21), 7426 (2020)

16. Kim, Y.: Convolutional neural networks for sentence classification. In: Moschitti, A., Pang, B., Daelemans, W. (eds.) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1746–1751. Association for Computational Linguistics, Doha, Qatar (Oct 2014). https://doi.org/10.3115/v1/D14-1181, https://aclanthology.org/D14-1181
17. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems **2**, 429–450 (2020)
18. Lin, B.Y., He, C., Ze, Z., Wang, H., Hua, Y., Dupuy, C., Gupta, R., Soltanolkotabi, M., Ren, X., Avestimehr, S.: FedNLP: Benchmarking federated learning methods for natural language processing tasks. In: Carpuat, M., de Marneffe, M.C., Meza Ruiz, I.V. (eds.) Findings of the Association for Computational Linguistics: NAACL 2022. pp. 157–175. Association for Computational Linguistics, Seattle, United States (Jul 2022). https://doi.org/10.18653/v1/2022.findings-naacl.13, https://aclanthology.org/2022.findings-naacl.13
19. Liu, Y., Fan, T., Chen, T., Xu, Q., Yang, Q.: Fate: an industrial grade platform for collaborative learning with data protection. J. Mach. Learn. Res. **22**(1) (jan 2021)
20. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)
21. Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., McMahan, H.B.: Adaptive federated optimization. arXiv preprint arXiv:2003.00295 (2020)
22. Reddi, S.J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., McMahan, H.B.: Adaptive federated optimization. CoRR **abs/2003.00295** (2020), https://arxiv.org/abs/2003.00295
23. Roth, H., Xu, Z., Renduchintala, A.: Adapting llms to downstream tasks using federated learning on distributed datasets. nvidia technical blog. Available at https://developer.nvidia.com/blog/adapting-llms-to-downstream-tasks-using-federated-learning-on-distributed-datasets/, last accessed 2024/02/13 (July 2023)
24. Schumann, G., Awick, J.P., Gómez, J.M.: Natural language processing using federated learning: A structured literature review. In: 2023 IEEE International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings). pp. 1–7 (2023). https://doi.org/10.1109/AIBThings58340.2023.10292481
25. Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D.S., Khazaeni, Y.: Federated learning with matched averaging. CoRR **abs/2002.06440** (2020), https://arxiv.org/abs/2002.06440
26. Ziller, A., Trask, A., Lopardo, A., Szymkow, B., Wagner, B., Bluemke, E., Nounahon, J.M., Passerat-Palmbach, J., Prakash, K., Rose, N., Ryffel, T., Reza, Z.N., Kaissis, G.: Pysyft: A library for easy federated learning (2021), https://api.semanticscholar.org/CorpusID:236690571