Exploring Pathways to Federated Learning for Intent Detection Advancements

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

[1] Tilde, Vienības gatve 75A, Riga, Latvia
{daiga.deksne,raivis.skadins}@tilde.com, jurgita.dzikiene@tilde.lt
[2] Faculty of Computing, University of Latvia, Raiņa bulv. 29, Riga, Latvia
[3] Faculty of Informatics, Vytautas Magnus University, Universiteto str. 10, Kaunas, Lithuania

Abstract. The abstract should briefly summarize the contents of the paper in 150–250 words.

Keywords: Intent Detection · Federated Learning · Virtual Assistants.

1 Introduction

In an age of digital communication, virtual assistants are essential for seamless interaction between organizations and stakeholders, addressing inquiries, providing round-the-clock support, and enhancing operational efficiency.

However, the development of reliable and versatile virtual assistants poses challenges, particularly in the realm of Natural Language Understanding (NLU). Understanding user intents accurately and responding adequately is fundamental to the effectiveness of virtual assistants, requiring sophisticated NLU models capable of interpreting diverse linguistic inputs.

This project is dedicated to tackling the complexities associated with NLU model development for virtual assistants, with a specific focus on possibilities to apply federated learning approaches, aiming to address key concerns such as data privacy.

Our objective is twofold: to construct independent bots tailored to the unique requirements of individual organizations, while concurrently creating a unified bot capable of serving the collective needs of all participating entities. This approach allows for bot development, ensuring that each organization's specific demands are met while maintaining a cohesive user experience across the board. Through a combination of innovative intent detection methods, advanced embedding models, and federated learning frameworks, we seek to achieve better results in the development of virtual assistants.

In this paper, we present the methodology, experiments, results, and conclusions of our project, offering insights into the potential of federated learning in NLU model development and its implications for the future of virtual assistants.

In addition to the advancements in virtual assistant technology, our project has made significant contributions to the field through the implementation of a federated learning-based intent detector. This novel approach, now available as open-source software, advances intent detection by leveraging federated learning techniques to enhance accuracy while maintaining data privacy and security. Furthermore, we have developed a comprehensive prototype that showcases the integration of this framework into RASA-based virtual assistants, including popular platforms like Bürokratt. This prototype not only demonstrates the efficacy of our approach but also provides practical guidance for incorporating federated learning into existing virtual assistant systems, thereby enabling organizations to deploy more intelligent and privacy-preserving conversational agents.

The structure of this document is the following. In Section 2 we outline the solution architecture. In Section 3 we describe development of the open-source software for federated training of intent detection models and implementation of the prototype. Section 4 delves into the research conducted within the project, covering the datasets utilized, the diverse intent detection methods tested, and the evaluation results obtained. Recommendations are provided regarding the most promising methods for practical implementation. Finally, the document concludes with a discussion on related work, contextualizing our implemented solution within the broader landscape of federated learning.

2 Related work

This chapter provides a summary of existing research on federated learning, covering various tasks, algorithms, challenges, and datasets commonly utilized in this domain. Our study specifically targets federated learning for intent detection, which involves addressing text classification problems. The main reasons federated learning is utilized in machine learning include:

- **Efficiency and Scalability**; By distributing computation to local devices, it reduces data transmission costs and eases the burden on central servers. This approach enables the training of models on large-scale datasets distributed across numerous devices or locations, thereby enhancing scalability. While federated learning historically addressed computing challenges, modern hardware has largely mitigated this issue for many machine learning tasks. However, in the era of Large Language Model (LLM) training, where both models and datasets are extensive, training efficiency becomes crucial. Frameworks such as FATE-LLM, FedML, and others utilize federated learning for LLM training, reflecting the evolving landscape of machine learning practices ([5], [9], [3], [26]).
- **Customization and Edge Computing**; Federated learning facilitates personalized model training, as models that are centrally pre-trained can be tailored to individual users' preferences or characteristics without compromising data privacy. It supports edge computing scenarios by enabling model fine-tuning directly on edge devices (such as mobile devices or IoT devices), reducing latency and improving real-time inference capabilities.

– **Privacy Preservation, Data Sovereignty, Security and Regulatory Compliance**; Federated learning allows training models on decentralized data sources without the need to share raw data, thus safeguarding privacy. Organizations can retain control over their data, preventing the need to share sensitive information with external parties. Federated learning frameworks help organizations comply with data protection regulations by minimizing data sharing and ensuring data privacy. Federated learning minimizes the risk of data breaches or unauthorized access, as data remains decentralized and is not stored centrally.

This research focuses primarily on addressing one aspect of the federated learning approach, namely privacy, security, or data confidentiality. Therefore, methods emphasizing training data distribution across large infrastructures or continuous bidirectional parameter synchronization between central and remote nodes are not the primary focus of this study. Federated learning finds application across various tasks and domains in machine learning. Here are some of the main tasks and domains where federated learning is commonly employed:

– Image Processing, Optical Character Recognition and Handwriting Recognition, where it is employed for various purposes including image classification, object detection, face recognition, image segmentation, and recognition of text within images or documents. These applications are commonly used for digitizing printed or handwritten text, as well as text extraction from digital photos. Typically, federated learning research related to image processing utilizes datasets such as the following: MNIST handwriting digit database [15], CIFAR-10/-100 Labeled small images [13], SVHN for number recognition [23], FashionMNIST Images of fashion products [31], Not-MNIST dataset of letters [2], FaceScrub face recognition dataset [24] and Traffic Signs dataset [27].
– Natural Language Processing (NLP), where federated learning is utilized across various tasks. These include text classification, where it is applied in sentiment analysis, intent detection, and spam detection. Federated learning also plays a role in Named Entity Recognition (NER), facilitating the identification of entities like names, dates, and locations within text documents, and federated learning is employed in Machine Translation tasks, aiding in the training of models to translate text from one language to another.
– Speech Recognition, where federated learning is employed to train models capable of transcribing spoken language into text. This technology is particularly relevant in virtual assistants and voice-controlled devices, where customization to the user's voice is crucial, all while maintaining strict user privacy and ensuring that private data remains on the device.
– Healthcare applications, where federated learning aids in disease prediction by leveraging patient data to forecast diseases or health outcomes while maintaining privacy. Additionally, federated learning plays a crucial role in medical image analysis, facilitating the examination of medical images such as X-rays and MRIs for diagnosis and treatment planning, while ensuring the confidentiality of sensitive medical information.

- Finance, where federated learning is employed in fraud detection, where it plays a critical role in identifying fraudulent transactions while safeguarding sensitive financial data. Additionally, federated learning is utilized in credit scoring, facilitating the assessment of credit risk based on customer data without the need to share personal information, thereby ensuring privacy and data confidentiality.
- and other applications include the Internet of Things (IoT), where it is employed to analyze data collected from IoT devices while ensuring user privacy is maintained throughout the process; in edge computing scenarios, where it enables model training directly on edge devices, thereby reducing latency and conserving bandwidth; and in autonomous vehicles for tasks such as object detection and lane detection, leveraging data from distributed vehicles to train models effectively.

This study primarily focuses on addressing a task within natural language processing, specifically intent detection. While intent detection may not be the most conventional application of federated learning in NLP, the majority of research is conducted on text classification, clustering, sentiment analysis, and recommendations. In these areas, classical datasets commonly used include the 20 newsgroup dataset [14], Reuters-21578 Text Categorization Collection [17], AG News [32], IMDb Reviews [20], Amazon Reviews [21], TREC [28], DBpedia [16], and others.

[19] offer a comprehensive exploration of federated learning within the realm of NLP. They argue that mounting concerns and regulatory frameworks surrounding data privacy and sparsity underscore the necessity for privacy-preserving, decentralized learning methods in NLP tasks. Federated learning emerges as a promising approach, enabling numerous clients such as personal devices or organizations to collaboratively learn a shared global model, thereby benefiting all participants while allowing individual users to maintain control over their data locally. The authors note a lack of systematic comparison and analysis in existing literature and introduce FedNLP, a benchmarking framework designed to evaluate federated learning methods across four distinct tasks, including text classification, which bears relevance to intent detection.

Classical algorithms in federated learning encompass diverse approaches aimed at facilitating collaborative model training across distributed clients while safeguarding data privacy. FedAvg [22] serves as the foundational method, assuming both clients and servers utilize the SGD optimizer for updating model weights. FedProx [18] addresses statistical heterogeneity by constraining local model updates closer to the initial global model through L2 regularization, enhancing training stability. FedOPT [25] extends the capabilities of FedAvg by introducing federated versions of adaptive optimizers such as Adagrad, Adam, and Yogi. These algorithms represent crucial advancements in federated learning research, providing essential tools for scalable and privacy-preserving model training across decentralized networks.

We conducted an analysis of various federated learning frameworks to explore their architectures, feature sets, and potential applications in our project. The

SEFT framework [29] offers features typical of federated learning solutions with one central node and multiple client nodes, focusing on efficiency and scalability in model training. Notably, it emphasizes efficient cryptography and robustness, making it resilient to client dropouts. Another framework, FLUTE (Federated Learning Utilities for Testing and Experimentation) [7], serves as a platform for high-performance federated learning simulations, addressing efficiency and scalability challenges in model training. FLUTE provides various features, enabling large-scale simulations with millions of clients and support for both single and multi-GPU setups, along with multi-node orchestration. Additionally, users have the flexibility to choose between local or global differential privacy and utilize model quantization techniques. [30] introduced a local differential privacy (LDP) based federated learning framework, catering to personalized privacy requirements of clients and mitigating threats posed by attackers attempting to infer clients' privacy through local model analysis. This framework primarily focuses on model customization and edge computing aspects in federated learning, particularly for image processing models. FEDML [9] stands as another ML library for large-scale distributed training, model serving, and federated learning. FEDML Launch, a cross-cloud scheduler, further enhances its capabilities, allowing for the execution of AI jobs on any GPU cloud or on-premise cluster.

3 Solution overview

This project is dedicated to addressing challenges associated with the development of sophisticated bots designed to cater to the diverse needs of multiple organizations. Our objective involves constructing independent bots for each organization while concurrently creating a unified bot capable of serving the collective requirements of all participating entities. This approach empowers us to meticulously develop and assess bots tailored to the unique demands of individual organizations. Subsequently, these individualized bots can seamlessly integrate into a unified bot, ensuring a cohesive user experience.

The rationale behind the creation of a unified bot stems from the realization that end users often lack awareness of, and interest in, the specific bot they are interacting with.

The solution architecture (Figure 1) encompasses various remote bot training sites where bot trainers autonomously develop and test their respective bots. These trainers manage their private training data, training local NLU models that can be employed within their specific remote bots as needed.

In addition to the remote training sites, a central training site plays a pivotal role. At this central location, a singular NLU model is trained using a federated training approach. The central federated training process aggregates NLU model parameters from the remote training sites, consolidating them into a cohesive federated NLU model. Notably, the training process not only acquires parameters from the remote nodes but may also incorporate its own training data. The outcome of this training process is a central bot equipped with the federated NLU model. This central bot possesses the capability to identify intents irre-

spective of their origin, as the federated NLU model recognizes intents defined in shared training data and any of the remote sites. This innovative approach ensures a versatile and inclusive bot system that effectively addresses the intricate needs of diverse organizations. The process of training intent detection models

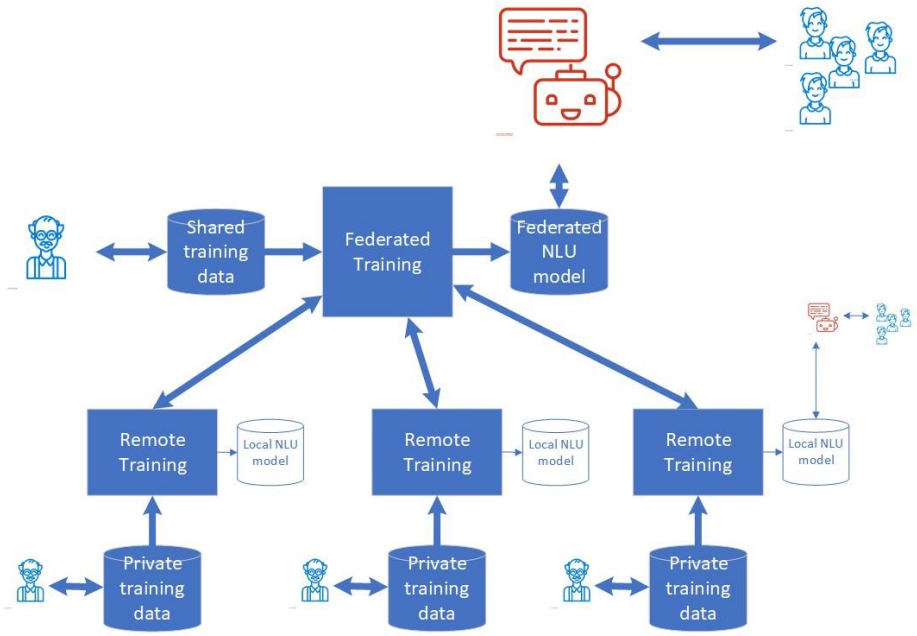

Fig. 1. System overview

traditionally involves utilizing a substantial and representative dataset. However, when dealing with data from diverse organizations, it becomes imperative to acknowledge potential sensitivities, with data owners being understandably hesitant to disclose the content of their data due to various reasons such as data quality, personal or classified information, or sensitive user queries posed to the bot.

The federated learning approach implemented in this project offers an effective solution to this privacy challenge. By employing federated learning, the central federated NLU model is trained without the need for sensitive data to leave the remote training sites. At no point does the central federated training process directly access or utilize the training data. Instead, the training data is vectorized on the remote site, and these vectors are utilized to compute the parameters of the local NLU model. Crucially, the raw data itself is never stored within the model.

Upon completion of the federated training process, only the model parameters and vectors are disclosed to the central training site. This approach ensures

that the data, in its textual form, remains securely within the premises of the data holder. Only binary representations of parameters are shared, safeguarding the privacy of the data, and significantly reducing the load associated with data exchange. This not only mitigates privacy concerns but also optimizes the efficiency of the overall federated learning system.

4 Solution implementation

The software for federated training and intent detection has been successfully implemented and is accessible on the GitHub repository: `https://github.com/tilde-nlp/fnlu`. This open-source software is made available under the Apache 2.0 license, encouraging collaborative use and contribution.

The main part of the solution is the federated intent detector. It's code serves a dual purpose, being utilized both in the training of intent detection models during the federated learning process and for real-time inference, specifically in runtime intent detection.

To configure the Federated NLU service, you have the option to choose between two different vectorization services employing LaBSE and SONAR embedding models. Vectorization services are distributed as Docker containers.

The product prototype developed in this project comprises two components. Firstly, there is an example implementation demonstrating how federated learning can be incorporated into Rasa bot software. The example provides guidance on integrating federated learning into the original Rasa source code, and a similar approach can be employed for integration into other software products based on Rasa technology, such as Bürokratt.

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

5 Intent detection

Many advanced embedding models support several languages including Estonian, and their linguistic knowledge includes vocabulary and sentence structures. In the semantic space, related or similar words are found close to each other, while unrelated words are further away from each other. Some embedding models extend this capability at the sentence level: sentences can be projected into the semantic space according to their similarity. Embedding models that understand languages and their relationships remain to be adapted to specific downstream tasks, e.g., by training classification models on top of them. In our case, the classification model is the intent detection model used as the main natural language understanding component in chatbots.

However, it's crucial to acknowledge that the availability of training data varies across languages, leading to an imbalance in the datasets used for train-

ing or fine-tuning embedding models for different languages. When training intent detection models, it is important to assess how all that multi-lingualism in embedding models affects it. It may result in lower accuracy for certain less-resourced languages.

5.1 Dataset

The intent detection research aimed to test the effectiveness of various text classification approaches for different languages and compare their results at the same time checking the possibilities of integrating other languages into the model in the future. Consequently, the experiments were performed with two datasets:

- *Multi-language segregated dataset* [4] (for the statistics see Table 1) containing several languages (English, German, French, Italian, Spanish, and Latvian). It 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. The texts were shuffled and split into training (80%) and testing (20%) subsets within 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) [5] were constructed using data provided by RIA from the Bürokratt project maybe we can add some reference here into that project. It would be nice to add similar table as Table 1: random, majority baselines, number of intents, is it also the single-label dataset?

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

, where P_i is the probability of the class.

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

5.2 Approaches

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

- **FastText+CNN**. The FastText [10] with Convolutional Neural Network (CNN) [12] approach involves using FastText embeddings and applying a refined Convolutional Neural Network architecture on top of these embeddings.

[4] 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.

[5] 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 1. Statistics of the multi-language segregated dataset.

Language	**Examples in training split**	**Examples 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

Table 2. Statistics of the Estonian datasets.

Dataset	**Number of intents**	**Number of examples**
Rahvusraamatukogu (National Library)	36	1 104
Sotsiaalkindlustusamet (Social Insurance)	7	79
Kriisijuhtimine (Crisis Management)	23	287

FastText makes use of subword embedding information to construct word embeddings, enabling it to understand the meaning of words even when they are not present in the vocabulary or contain spelling errors. This capability is especially valuable for addressing typos and out-of-vocabulary words that were not present in the training data. CNNs are robust at identifying local patterns and characteristics within textual data. When combined with FastText embeddings, they can effectively extract valuable features from text, including token n-grams or their convolutions, which play a crucial role in text classification tasks where keywords and text snippets are especially important. In this research, we have used the architecture of the CNN method presented in [11]. This method will exclusively serve as the baseline.

– **LaBSE+FFNN**. We utilize the frozen Language Agnostic BERT Sentence Embedding (LaBSE) approach, as introduced by [6], along with a straightforward two-layer fully connected (FFNN) fine-tuned for our classification tasks. In contrast to traditional BERT embeddings that offer representations at the word level, LaBSE specializes in generating embeddings at the sentence level, capturing the semantics of the entire text simultaneously. LaBSE is a multilingual model that can accommodate 109 different languages (including Estonian). LaBSE generates a fixed-size vector for the entire text, meaning it doesn't retain word boundaries, as is the case with BERT. Since LaBSE doesn't consider word order in sentences, it is particularly well-suited for languages with flexible sentence structures, requiring less extensive training data to account for various structural variations. However, LaBSE doesn't provide equal coverage for all languages [6]. Consequently, the cross-lingual capability may not be equally strong across all languages.

- **LaBSE-fine-tuning**. This method closely resembles LaBSE+FFNN, but instead of maintaining the LaBSE part with fixed , all layers are unfrozen, and all parameters are adjusted during fine-tuning together with the additional fully connected layer mapping LaBSE to the intents. In the LaBSE+FFNN approach, there are around 1 thousand parameters to be learned during training. Whereas, in LaBSE-fine-tuning, the number of parameters to be adjusted is approximately 0.5 billion, leading to a substantial increase in computational and time requirements.
- **LaBSE-LangChain-k1**. It utilizes the LangChain framework, which allows for the creation of context-aware applications that invoke the power of language models. No training is required; this approach exclusively involves vectorizing the training instances and retrieving the most similar ones to the test example using the computed cosine similarity value. In this specific method, we employ LaBSE sentence embeddings and a greedy search to find the training instance that is most similar to the test instance. The test instance is then assigned the label of the nearest training instance.
- **LaBSE-LangChain-k10-mv**. This method closely resembles LaBSE-LangChain-k1, but instead of searching for the single nearest 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 is the same as LaBSE-LangChain-k1, but instead of LaBSE embeddings, OpenAI's text-embedding-ada-002 embeddings [8] are used 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 used OpenAI's davinci-002 model [1]. This model is a generative transformer model, and we adjusted it to generate only one first token as the input text's label (intent). In our experiments, we fine-tuned the added layers' parameters while keeping its hyperparameter values (learning rate, number of added layers, etc.) at their defaults.

5.3 Experiments and Results

During the experimental investigation, first we applied the approaches previously introduced to the non-Estonian dataset. For the methods that required training (FastText+CNN, LaBSE+FFNN, LaBSE-fine-tuning, and Davinci-fine-tuning), the training dataset was shuffled and split into the training (80%) and validation (20%) sets. The accuracy metric was selected as the main evaluation metric: correctly-labelled-testing-instances/all-testing-instances.

We conducted five test runs for the LaBSE+FFNN and LaBSE-fine-tuning methods to account for randomness in their parameter initialization. We averaged these results and calculated confidence intervals accordingly. Similarity-based approaches do not have randomness in their evaluation procedure, always

resulting in the same prediction. On the contrary, FastText+CNN and Davinci-fine-tuning indeed have randomness in their training mechanisms. However, the first solution serves as the baseline approach; therefore, it does not necessitate a thorough examination. The Davinci-fine-tuning was tested once just for comparison purposes. This approach is not suitable as all training data and the fine-tuned model are stored on third-party servers which is not compatible with our project requirements, besides fine-tuning the model and its subsequent use incur charges. Figure 2 depicts the outcomes of all tested approaches. The English

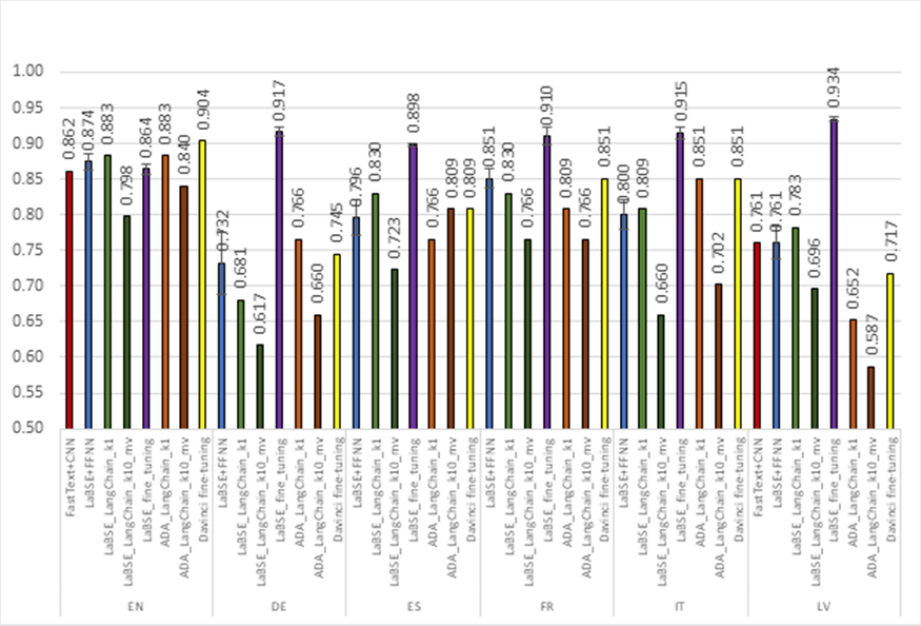

Fig. 2. The accuracy values with different classification, similarity-based, and generative approaches across various languages

dataset yielded the most favorable results with Davinci-fine-tuning, followed by LaBSE-LangChain-k1, ADA-LangChain-k1, and LaBSE-fine-tuning. In contrast, LaBSE-fine-tuning emerged as the most effective choice for all other languages (German, Spanish, French, Italian, and Latvian).

Despite its superior performance, LaBSE-fine-tuning is deemed impractical for this project due to its extensive time requirements (up to 1 hour for training) and high hardware demands (exceeding 12 GiB of GPU RAM). Such resource-intensive processing is not feasible for practical applications, given the preference for shorter training times, even 15 minutes being considered lengthy for bot trainers. Consequently, the next viable option is LaBSE-LangChain-k1, which delivers commendable results for all non-English languages except German. No-

tably, ADA and Davinci models outperform LaBSE models for German, likely owing to the rich representation of German in these models. For further experiments with Estonian, LaBSE-LangChain-k1 is selected as the primary method. Subsequently, SONAR-LangChain-k1, a similar method utilizing the SONAR language model [4] from Meta Research instead of LaBSE, is introduced to expand the scope of experimentation, as SONAR is lately getting attention in many text classification related tasks. This strategic selection aims to balance performance and practicality in the context of the project's objectives.

As Estonian as opposed to English, German, etc. is a less-resourced language. Available text data and computing resources might be insufficient to train the high-quality language model for the Estonian. Our chosen Intent detection technique relies on the multi-lingual large language models for computation of sentence embeddings. In scope of the project, we have experimented with LaBSE and SONAR multi-lingual language models. Results of the experiments are summarized in Table 3. Results with SONAR are slightly better. On the other hand, LaBSE is faster. LaBSE takes 27-30 ms per request, SONAR – 120-130 ms per request. When loaded in memory both models take 2-3 GiB of RAM. LaBSE model uses 5.27 GiB of disk space, while SONAR model uses 2.85 GiB of disk space. In these experiments, the intent detector functions as a classifier that

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

identifies the most probable intents by assessing the semantic similarity of sentence embedding vectors derived from the Large Language Model (LLM). The embedding vectors of the training data, along with intent IDs and metadata about the source, are stored in the FAISS vector store. FAISS is equipped with algorithms for efficient similarity search and clustering of dense vectors.

During inference, the FAISS index is queried to identify entries with embedding vectors most similar to the user's sentence embeddings vector. The corre-

sponding intent IDs of these entries are then returned, along with confidence scores. The similarity of vectors is determined by their proximity in Euclidean space, providing an effective mechanism for intent detection based on semantic similarities.

5.4 Discussion and Conclusions

All used scenarios and approaches are suitable for our problem-solving, as achieved accuracies significantly outperform random and majority baselines.

The best results on the English dataset were achieved through Davinci-fine-tuning, followed by LaBSE-LangChain-k1, ADA-LangChain-k1, and then LaBSE-fine-tuning. Conversely, LaBSE-fine-tuning proved to be the most effective for all other languages (German, Spanish, French, Italian, and Latvian).

LaBSE-fine-tuning, despite superior performance, is impractical due to extended training times and high hardware demands. LaBSE-LangChain-k1 emerges as a viable alternative, delivering commendable results for non-English languages. For Estonian experiments, LaBSE-LangChain-k1 is chosen as the primary method for Estonian intent detection due to its significantly faster training and inference times compared to SONAR-LangChain-k1, despite the latter exhibiting slightly better quality. Certainly, the dataset used in our experiments is quite small. Consequently, evaluating differences and their statistical significance proves to be challenging. Nonetheless, this dataset is in line with our customers' usual expectations, and as this research demonstrated, it delivers favorable outcomes even when working with limited data.

In conclusion, the multi-language Language Models demonstrate effectiveness, particularly in languages lacking readily available NLP tools and resources. The implementation of FAISS algorithms enables rapid searches in the index, facilitating the merging of FAISS models. Notably, the federated learning approach employed allows for flexibility, as there is no need for a pre-defined set of intents during central model training. Each client can introduce new intents, modify existing ones, recompile local models, and update the central model, mitigating the parameter aggregation issues seen in standard Federated Learning (FL) algorithms. Furthermore, the system accommodates variations in the number of training samples and intents among users, and intentionally bad data does not compromise the central model integrity, as intent records in the index operate independently of each other.

6 Conclusions

The pursuit of developing sophisticated virtual assistants capable of meeting the diverse needs of multiple organizations has led us to explore innovative solutions in Natural Language Understanding. Our project aims to construct independent bots tailored to individual organizations while simultaneously creating a unified bot capable of serving the collective requirements of all participating entities.

This approach allows for meticulous bot development, ensuring that each organization's unique demands are met while maintaining a cohesive user experience across the board.

By adopting a federated learning approach, we have addressed significant challenges associated with privacy, scalability, and efficiency in NLU model training. The federated learning process, involving both remote and central training sites, enables the aggregation of NLU model parameters without the need for sensitive data to leave the remote sites. This ensures data privacy while optimizing training efficiency and scalability, crucial factors in developing reliable virtual assistants.

Our experiments with intent detection methods across various languages have yielded promising results. Leveraging advanced techniques such as FastText+CNN, LaBSE, and SONAR embeddings, we have achieved accuracies significantly surpassing evaluation baselines. While certain methods exhibit superior performance, practical considerations such as training time and hardware requirements must be taken into account. LaBSE vectorization model with vector similarity search emerges as a viable option, delivering commendable results across several languages including Estonian and demonstrating faster training and inference times compared to other methods.

Looking ahead, our research underscores the effectiveness of multi-language Language Models in NLU tasks, particularly in less-resourced languages like Estonian. The implementation of FAISS algorithms enables rapid searches in indexes, facilitating the integration of FAISS models into our federated learning system. Furthermore, our federated learning approach allows for flexibility and adaptability, accommodating variations in training data and intents among users while ensuring data privacy and model integrity.

In conclusion, our project represents a significant step forward in developing sophisticated virtual assistants through federated learning and advanced intent detection methods. By leveraging innovative techniques and frameworks, we have laid the foundation for a versatile, scalable, and privacy-preserving NLU system capable of meeting the diverse needs of multiple organizations in various linguistic contexts.

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

Disclosure of Interests. It is now necessary to declare any competing interests or to specifically state that the authors have no competing interests. Please place the statement with a bold run-in heading in small font size beneath the (optional) acknowledgments[6], for example: The authors have no competing interests to declare that are relevant to the content of this article. Or: Author A has received research grants from Company W. Author B has received a speaker honorarium from Company X and owns stock in Company Y. Author C is a member of committee Z.

[6] If EquinOCS, our proceedings submission system, is used, then the disclaimer can be provided directly in the system.

References

1. 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)
2. Bulatov, Y.: Notmnist dataset. google (books/ocr). Google (Books/OCR) (2011)
3. Chen, C., Feng, X., Zhou, J., Yin, J., Zheng, X.: Federated large language model: A position paper (2023)
4. Duquenne, P.A., Schwenk, H., Sagot, B.: Sonar: Sentence-level multimodal and language-agnostic representations (2023)
5. 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)
6. 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
7. 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)
8. 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)
9. He, C., Li, S., So, J., Zeng, X., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H., Zhu, X., Wang, J., Shen, L., Zhao, P., Kang, Y., Liu, Y., Raskar, R., Yang, Q., Annavaram, M., Avestimehr, S.: Fedml: A research library and benchmark for federated machine learning (2020)
10. 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
11. Kapočiūtė-Dzikienė, J., Balodis, K., Skadiņš, R.: Intent detection problem solving via automatic dnn hyperparameter optimization. Applied Sciences **10**(21), 7426 (2020)
12. 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
13. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto (2009)
14. Lang, K.: Newsweeder: Learning to filter netnews. In: Machine learning proceedings 1995, pp. 331–339. Elsevier (1995)
15. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
16. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al.: Dbpedia–a large-scale,

multilingual knowledge base extracted from wikipedia. Semantic web **6**(2), 167–195 (2015)
17. Lewis, D.: Reuters-21578 text categorization test collection. Distribution 1.0, AT&T Labs-Research (1997)
18. 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)
19. 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
20. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Lin, D., Matsumoto, Y., Mihalcea, R. (eds.) Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. pp. 142–150. Association for Computational Linguistics, Portland, Oregon, USA (Jun 2011), https://aclanthology.org/P11-1015
21. McAuley, J., Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In: Proceedings of the 7th ACM conference on Recommender systems. pp. 165–172 (2013)
22. 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)
23. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011 (2011), http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf
24. Ng, H.W., Winkler, S.: A data-driven approach to cleaning large face datasets. In: 2014 IEEE international conference on image processing (ICIP). pp. 343–347. IEEE (2014)
25. 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)
26. 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)
27. Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: The german traffic sign recognition benchmark: a multi-class classification competition. In: The 2011 international joint conference on neural networks. pp. 1453–1460. IEEE (2011)
28. Voorhees, E.M., Harman, D.K., et al.: TREC: Experiment and evaluation in information retrieval, vol. 63. MIT press Cambridge (2005)
29. Wang, C., Deng, J., Meng, X., Wang, Y., Li, J., Lin, S., Han, S., Miao, F., Rajasekaran, S., Ding, C.: A secure and efficient federated learning framework for nlp. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 7676–7682 (2021)

30. Wu, X., Xu, L., Zhu, L.: Local differential privacy-based federated learning under personalized settings. Applied Sciences **13**(7), 4168 (2023)
31. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms (2017)
32. Zhang, X., LeCun, Y.: Text understanding from scratch (2015)