iASSIST: Low-cost, portable and embedded assistants for on-premise automated transcription and translation services

iASSIST: Asistentes embebidos, portables y de bajo coste para servicios on-premise de transcripción y traducción

Aitor Álvarez[1], Víctor Ruiz[1], Iván G. Torre[1], Thierry Etchegoyhen[1], Harritxu Gete[1] and Joaquín Arellano[1]

[1]Fundación Vicomtech, Basque Research and Technology Alliance (BRTA), Donostia-San Sebastián, 20009, Spain

Abstract

We present iASSIST, a low-cost, portable and embedded solution for on-premise automated neural transcription and translation services, currently for the English, Spanish and Basque languages. The system is fully operational, embedded in Jetson boards, and accessible via a user-friendly interface to perform real-time transcription and translation with high-quality neural models.

Keywords

edge computing, embedded AI, neural transcription, neural translation

1. Introduction

Recent advances in deep neural networks (DNNs) have led to significant improvements in both Automatic Speech Recognition (ASR) and Neural Machine Translation (NMT) [1, 2]. However, these advances are mainly achieved with large neural architectures, trained on massive volumes of data and typically deployed on high-end expensive servers in the cloud to provide efficient services, which raises a number of critical issues.

First, privacy is an important concern, since sending personal or confidential data over the Internet makes the information vulnerable to attacks and breaches. The General Data Protection Regulation (GDPR) and similar policies set to protect sensitive data also need to be taken into account.

Secondly, high-quality AI models typically require servers with significant computational capacity and GPU acceleration cards for both training and inference. Acquiring this type of hardware resources for local computing, or renting appropriate infrastructure in the cloud, can represent a significant budget that many companies cannot cover.

Thirdly, deep AI models are significantly impacting energy consumption worldwide, with serious consequences on the increasing climate crisis. Reducing the ecological footprint of current AI technology is a critical part of the current research agenda.

Finally, latency issues and information loss can impact cloud computing services, making it difficult at times to deploy responsive and robust AI solutions.

Edge computing aims to move computational power and data processing closer the originating data [3], with AI algorithms running on local networks or embedded devices to guarantee data privacy and reduce latency, energy consumption and network load. However, integrating high-performance AI models into embedded systems with low computational capabilities requires system and model optimization.

Within this context, we present iASSIST, a low-cost, portable and embedded solution for on-premise automated neural transcription and translation services for the English, Spanish and Basque languages. This solution has been developed within the applied research project iASSIST, partially supported by the Department of Economic Development of the Basque Government. The project started in September 2019 and finalised in December 2021, and

SEPLN-PD 2022. Annual Conference of the Spanish Association for Natural Language Processing 2022: Projects and Demonstrations, September 21-23, 2022, A Coruña, Spain

aalvarez@vicomtech.org (A. Álvarez); vruiz@vicomtech.org (V. Ruiz); igonzalez@vicomtech.org (I. G. Torre); tetchegoyhen@vicomtech.org (T. Etchegoyhen); hgete@vicomtech.org (H. Gete); jarellano@vicomtech.org (J. Arellano)

0000-0002-7938-4486 (A. Álvarez); 0000-0003-2380-010X (I. G. Torre)

was carried out by the following consortium: SPC[1] (project coordinator), MondragonLingua[2], Serikat[3], Natural Vox[4], Haresi[5] and Vicomtech[6].

2. iASSIST

The core architecture of iASSIST is shown in Figure 1. It consists of the following main components:

- A front-end, composed of a web-based graphical user interface (GUI).
- A REST API, which exposes the functionalities of the back-end.
- A back-end, which orchestrates all the functionalities of the solution, including automatic transcription and translation, client request management, model loading and unloading, and operational modes (batch and streaming).

Among the different options for embedded systems offered by the market (e.g. Raspberry Pi, NVIDIA Jetson, Google Coral or Intel Movidius, among others), we selected the NVIDIA Jetson embedded computing boards for the project. Specifically, we focused on two specific devices with different capabilities: Jetson TX2 and Jetson AGX Xavier. Although these two boards were relatively similar prices at the time, the AGX Xavier (32 TOPS, 512-core GPU, 8-core CPU, 32 GB of shared memory) offered significantly more computational power than the TX2 system (1.3 TOPS, 256-core GPU, dual-core CPU, 8 GB of shared memory), while also being more energy efficient. During the project, we explored the capacities of both boards and evaluated the integration of different AI models depending on their architecture, size, number of parameters and performance in each embedded system.

In the following subsections, each of the main components of the iASSIST solution is presented in more detail.

2.1. Front-end

The iASSIST GUI aims to facilitate the communication between the user and the back-end. It was designed from a usability and user experience perspective, prioritizing simplicity. The GUI provides users with different input options, from text to audio file (batch mode) and audio source (streaming mode), and allows them to select different transcription and translation models to perform the corresponding tasks. Additionally, it integrates two main text-boxes to present the transcription and translation results and a graphical interface to manage model loading and unloading in memory. It is worth noting that the transcription results can be downloaded in different formats (txt, rtf, xml, srt, vtt) that can be used for different applications such as subtitling, keyword spotting and rich transcription. The GUI was developed using the Angular framework[7] and deployed via a Nginx web server[8].

2.2. REST API

The REST API serves as the main interface between the GUI and the back-end. In addition, it provides an alternative way for the user to directly access all the features of the solution via http requests, allowing third party systems to be built on top of iASSIST and thus extend its functionality.

2.3. Back-end

The iASSIST back-end is composed of several modules which encompass the features of the solution. The main modules are described in turn in the next subsections.

2.3.1. Orchestrator

This module encompasses the automated configuration, management, and coordination of the main components and services of the back-end. At its core, it manages user requests, communication between modules and I/O interaction. The module also implements the logic and interfaces for the batch and streaming applications, manages automatic language identification for translation with bilingual ASR models, and controls the input sources, devices and audio streams. The iASSIST solution is able to process audio files, texts or streaming audio coming from any microphone connected to the board or machine where the GUI is launched.

2.3.2. Model management

Running applications composed of several AI models on embedded systems requires dynamically controlling model activation and memory usage, given

[1] https://www.spc.es/
[2] https://www.mondragonlingua.com/en
[3] https://www.serikat.es/
[4] https://www.naturalvox.eu/en/home/
[5] https://haresi.es/
[6] https://www.vicomtech.org/en
[7] https://angular.io/
[8] https://www.nginx.com/

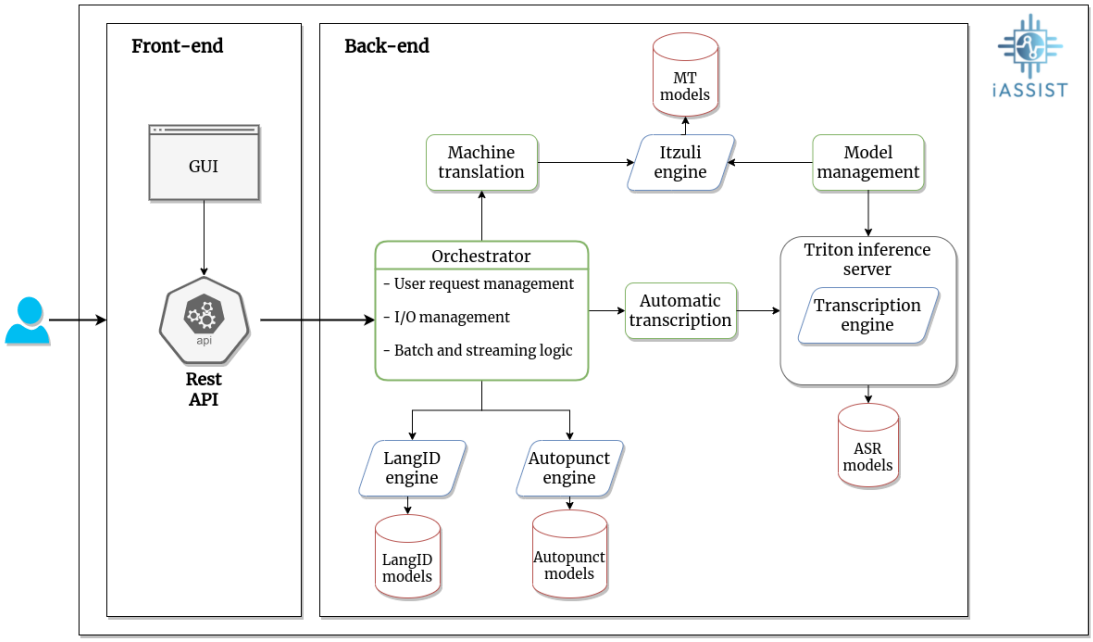

Figure 1: Core architecture of the iASSIST solution.

common limitations of the supporting boards. The model management module ensures proper model loading and unloading in memory, allowing users to enable or disable the relevant functionality depending on the AI task at hand.

2.3.3. Automatic transcription

The Automatic Transcription module is managed by the Triton Inference Server[9], which is in charge of handling workloads and integrating the three main modules of the transcription pipeline. The first module processes the raw audio input by extracting features as spectrogram chunks, which are sent to an acoustic model for probabilistic classification in a second stage. The final module, composed by the decoder, determines the most likely transcription for that audio using the likelihoods produced by the previous classification with the help of a language model.

For iASSIST, we developed acoustic models based on the NVIDIA's Quartznet E2E architecture [4], designed by the need to reduce the size and complexity of the recognition models, making them lighter, faster and more easily deployed on embedded systems. This architecture is composed of multiple blocks with residual connections in between. Each unique block consists of one or more modules with 1D time-channel separable convolutional layers, batch normalisation, and ReLU layers.

For each of the selected Jetson embedded systems, we experimented with different versions of the Quartznet architecture. After evaluating their performance in terms of latency and quality, we decided to deploy the Quartznet Q15×5 based model on the Jetson AGX Xavier, and the Q10×5 based model on the Jetson TX2 board. The main difference lies in the number of times the Quartznet models repeat the five unique blocks, which modifies the total number of parameters from 18.9M (Q15×5) to 12.8M (Q10×5). To further optimize the performance of the Quartznet acoustic models, quantization and layer fusion techniques were also applied via the TensorRT library [5].

Finally, the raw transcriptions are enriched with capitalisation and punctuation marks generated by the BERT-based AutoPunct engine [6]. In addition to enhancing readability, splitting the raw text into correctly punctuated sentences increases the quality of machine translation results.

[9]https://developer.nvidia.com/nvidia-triton-inference-server

2.4. Machine Translation

The Machine Translation module is based on Vicomtech's Itzuli Translator engine, a robust and scalable text translation system, which can be deployed under Kubernetes orchestration or as a standalone platform in a dedicated server, and integrates MarianNMT [7] in its own back-end to perform efficient NMT inference.

To optimise Transformer [2] NMT models, in terms of size and inference latency, we explored different strategies based on network pruning, quantization and knowledge distillation. Our final optimised models, suitable for the more constrained TX2, were student models trained on the knowledge distilled by large teacher models, with 6 Self-Attention layers for encoding and 2 SSRU layers [8] for decoding. The student models halved the memory footprint of teacher models, increased inference speed between 200% and 400% depending on beam size, with minor losses in terms of translation quality ranging between 0.2 and 1.4 BLEU points.

Translation models can be loaded and unloaded in memory on the fly, thus giving users the ability to switch to new translation tasks as needed within the constrained environment. Translation can be performed directly on user-provided source text or on the output of the ASR component to perform real-time speech translation.

3. Conclusions

We described the iASSIST solution, an embedded assistant for on-premise neural transcription and translation services. The application was validated by each of the companies of the consortium within three evaluation campaigns, where they accessed the embedded system externally and tested the solution at operational, usability and quality levels over their own contents and devices.

iASSIST demonstrates the ability to embed neural transcription and translation technology in Jetson boards with hardly any loss in performance, performing both batch and streaming tasks within a secure, portable and low-cost edge device. As future work, we will explore other embedded systems in which iASSIST could be integrated and will continue to improve AI model optimization for less powerful environments, particularly CPU-based client-side computation.

Acknowledgments

iASSIST is partially funded by the Basque Business Development Agency, SPRI, under grant agreement ZL-2021/00103.

References

[1] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, et al., State-of-the-art speech recognition with sequence-to-sequence models, in: Proc. of ICASSP, 2018, pp. 4774–4778.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 6000–6010.

[3] V. K. Sarker, J. P. Queralta, T. N. Gia, H. Tenhunen, T. Westerlund, A survey on LoRa for IoT: Integrating edge computing, in: Proc of FMEC 2019, 2019, pp. 295–300.

[4] S. Kriman, S. Beliaev, B. Ginsburg, J. Huang, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li, Y. Zhang, Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions, in: Proc. of ICASSP 2020, 2020, pp. 6124–6128.

[5] H. Vanholder, Efficient inference with TensorRT, in: GPU Technology Conference, volume 1, 2016, p. 2.

[6] A. González-Docasal, A. García-Pablos, H. Arzelus, A. Álvarez, AutoPunct: A BERT-based automatic punctuation and capitalisation system for Spanish and Basque, Proces. de Leng. Nat. 67 (2021) 59–68.

[7] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. Fikri Aji, N. Bogoychev, A. F. T. Martins, A. Birch, Marian: Fast neural machine translation in C++, in: Proc. of ACL 2018, 2018, pp. 116–121.

[8] Y. J. Kim, M. Junczys-Dowmunt, H. Hassan, A. F. Aji, K. Heafield, R. Grundkiewicz, N. Bogoychev, From research to production and back: Ludicrously fast neural machine translation, in: Proc. of WNGT, 2019, pp. 280–288.