MTUOC server: integrating several NMT and LLMs into professional translation workflows

Antoni Oliver
Universitat Oberta de Catalunya
aoliverg@uoc.edu

Abstract

In this paper, we present the latest version of MTUOC-server and MTUOC-multiserver, a robust tool capable of launching one or more translation servers. It supports a wide range of NMT systems and LLM models, both commercial and open-source, and is compatible with several communication protocols, broadening the range of tools it can work with. This server is a component of the MTUOC project and is distributed under a free license.

1 Introduction

The number of available machine translation (MT) systems has significantly increased in recent years, and with the successful integration of Large Language Models (LLMs) for translation, the variety of options is higher than ever. However, not all of these systems are suitable for professional translation environments, as they cannot be easily integrated with existing translation tools. While some of these systems offer impressive quality, they may lack essential features, such as the restoration of XML tags, which are crucial in real-world translation scenarios. Even more, the combination of several MT tools in a single workflow is not always straightforward.

Taking all these factors into account, and in line with objectives of the MTUOC (**M**achine **T**ranslation at **U**niversitat **O**berta de **C**atalunya) project[1], which aims to make advanced MT technologies more accessible to everyone, we have developed new versions of the MTUOC server and a companion program called MTUOC-multiserver. The MTUOC-server is a software application designed to interface with a single MT or LLM translation system, supporting multiple communication protocols to ensure compatibility with a wide range of client applications. It can integrate with an extensive array of translation services and tools, a capability that continues to expand with each new version. The MTUOC-multiserver is a software application capable of connecting with multiple MTUOC-servers, aggregating translation candidates from each server, and ranking them based on predefined metrics or criteria. This ensures that the top-ranked translation candidate is the best among all received options.

2 Main features

2.1 Multiplatform Support

The MTUOC-server is designed to maximize compatibility with major operating systems, including Linux, Windows, and macOS. However, running the translation service locally may not always be feasible, as some engines run only on specific operating systems.

2.2 Hardware requirements

The MTUOC server is designed to work on any computer provided it has enough memory to load the required models. No powerful servers are needed and the MT systems and LLMs models can be used in systems with no GPU available, with the consequent decrement of translation speed. This enables access to advanced translation systems for any user.

2.3 Implemented communication protocols

MTUOC can be configured to communicate with client applications using one of the following protocols: the MTUOC protocol (specific to the MTUOC project), Moses, ModernMT, OpenNMT, or NMTWizard.

[1] https://mtuoc.github.io/

Proceedings of Machine Translation Summit XX Volume 2, pages 81–82, June 23-27, 2025

2.4 Available MT systems and LLMs

The MTUOC-server currently supports a wide range of MT systems, encompassing both open-source and commercial options. Accessing commercial systems requires an API key. The currently supported MT systems include Marian, OpenNMT, Moses, OpusMT, NLLB, ctranslate2, Transformers, Aina, Softcatalà, Apertium, Google Translate, DeepL, and Lucy. We are now integrating LLMs for translation, with the following models slated for inclusion: Salamandra[2], both instruct and translation models; Bloom[3], as it has demonstrated good translation performance (Bawden and Yvon, 2023); ChatGPT, as it's being used for translation (Peng et al., 2023), DeepSeek, accessed either with the API or querying the full model downloaded from HuggingFace using the transformers library. When using LLMs the user can specify the prompt for the translation and, if needed, a regular expression to retrieve the translation from the answer. We plan to include additional NMT systems or LLM models as they become available in the future. The integration of new models is straightforward, as it can be achieved through the adaptation of the existing MTUOC modules, which are implemented as specific Python classes initialized when the MTUOC-server starts.

2.5 Restoration of XML tags

Some MT engines cannot accurately retrieve the positions of XML tags in the target segment that are present in the source. This limitation is critical when translating complex formats, such as DOCX files. The MTUOC-server includes a tag restoration algorithm that utilizes word (or subword) alignments. When the MT system provides these alignments, MTUOC uses them for tag restoration. For MT systems that do not supply this information, MTUOC can rely on external fast_align models to calculate the word alignments.

2.6 Reordering of candidates

Some MT and LLM models can provide a set of translation candidates ranked by an internal measure. Using the MTUOC-Multiserver, it is possible to retrieve multiple translations from various MT systems or LLMs. In both cases, the server can reorder these candidates based on external quality estimation metrics, such as SBERT cosine similarity or COMET, ensuring that the first candidate provided is the one with the highest value for the chosen metric.

2.7 Use of translation memories

MTUOC-server can integrate translation memories and return retrieved translations if the match score is higher than a predefined threshold. It can be configured to return either only the match or the match integrated into the MT candidates, positioned according to the match score.

2.8 Plugins for CAT Tools

We provide plugins for the following popular CAT tools: OmegaT[4] and RWS Trados Studio[5]. These plugins are designed to work with the MTUOC communication protocol. Additionally, if the MTUOC-server is started using a different protocol, it can also be compatible with other tools. For instance, starting the server with the ModernMT protocol enables seamless compatibility with Okapi tools such as Rainbow or Tikal. We also provide a desktop application, MTUOC-Translator[6], which can be used to translate documents using the MTUOC-server. Additionally, a web application[7] is available for translating documents through MTUOC-server.

3 Conclusions and future work

We have presented a tool that allows to integrate several MT and LLM into professional translation workflows. The tool holds a free license (GNU-GPL) and it can be downloaded from Github.[8] As a future work, we plan to explore more efficient ways to query LLMs, as ollama or llama.cpp.

References

Rachel Bawden and François Yvon. 2023. Investigating the translation performance of a large multilingual language model: the case of Bloom. *arXiv preprint arXiv:2303.01911*.

Keqin Peng, Liang Ding, Zhong, et al. 2023. Towards making the most of ChatGPT for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5622–5633.

[2]https://huggingface.co/collections/BSC-LT/
[3]https://huggingface.co/bigscience/bloom
[4]https://github.com/mtuoc/MTUOC-OmegaT-plugin
[5]https://github.com/mtuoc/MTUOC-Trados-plugin
[6]https://github.com/mtuoc/MTUOC-Translator
[7]https://github.com/mtuoc/MTUOC-web-translator
[8]https://github.com/mtuoc/MTUOC-server