nEYron: Implementation and Deployment of an MT System for a Large Audit & Consulting Corporation

Artur Nowakowski[1,2], **Krzysztof Jassem**[1,2], **Maciej Lison**[1], **Rafał Jaworski**[2], **Tomasz Dwojak**[2]
[1] Poleng, Poznań, Poland
{name.surname}@poleng.pl
[2] Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Poznań, Poland
{name.surname}@amu.edu.pl

Karolina Wiater, Olga Posesor
EY Poland, Warsaw, Poland
{name.surname}@pl.ey.com

Abstract

This paper reports on the implementation and deployment of an MT system in the Polish branch of EY Global Limited. The system supports standard CAT and MT functionalities such as translation memory fuzzy search, document translation and post-editing, and meets less common, customer-specific expectations. The deployment began in August 2018 with a Proof-of-Concept, and ended with the signing of the Final Version acceptance certificate in October 2021. We present the challenges that were faced during the deployment, particularly in relation to the security check and installation processes in the production environment.

1 Business Need

On March 6, 2018, the Polish parliament adopted a law that laid down rules for the Polish Agency of Audit Surveillance regarding the control of auditing companies. The law states that "Documents presented by the audited company for the needs of the surveillance are drawn up in Polish or the audit company provides their translation into Polish." The law forced auditing companies to provide Polish translations for large volumes of English texts. That triggered the idea, at the Polish branch of EY Global Limited (EY Poland), that the cost of the task might be reduced if it were assisted by a translation engine. EY Poland contacted the company Poleng Ltd. (Poleng) to verify the possibility of using their product, TranslAide Workspace, for the task. During initial discussions, EY Poland came to the conclusion that it might be beneficial for the company to have the software installed and running on site.

2 The Story of the Deployment

2.1 TranslAide Workspace

The first phase of the deployment began in August 2018. The deployed system was based on TranslAide Workspace, which combined computer-aided translation (translation memory with fuzzy search and segment-by-segment editing) with a generic machine translation engine, not trained specifically on the in-domain data. The task consisted in replacing the existing translation engine with a new one, dedicated to the customer.

The deployment was divided into the Proof-of-Concept (POC) and Final Version stages. The POC machine was to be installed in the Linux environment to make the initial deployment easier for the Poleng team. There were no explicit expectations regarding the quality of the translation imposed on the POC version. However, moving forward to the Final Version stage was conditional on acceptance of the POC by the customer – including translation quality, which would be checked by human specialists from the EY corporation. The Final Version – all of the system components, including model training – was expected to run on the Windows operating system to meet EY's security standards and internal regulations.

The expectations for the system were the following: The TranslAide Workspace system would consist of three modules – Web Application, Translation Memory, and Machine Translation Service:

- Web Application would be the part of the sys-

tem with which the user interacts;

- Translation Memory would provide translation of segments that were found in its database;
- Machine Translation Service would provide translation of all remaining sentences at a speed not slower than a second per segment.

(Details on current expectations for the three modules are given in section 4.)

All system components, as well as the training of the models, should be run on a PC machine with the following specification: NVIDIA GTX 1080Ti GPU, 32 GB RAM and an 8-core processor.

The POC phase ended on schedule (within three months), but the translation quality was not fully satisfactory, as the system sporadically produced incorrect translations of some acronyms and rare words; the issue resulted from certain flaws in subword handling by Marian NMT (Junczys-Dowmunt et al., 2018). On rare occasions, the system would also crash when importing a PowerPoint presentation, because of improper handling of some XML tags specific to the PowerPoint document's internal structure. After the major issues had been identified and fixed, the Final Version was developed for the Windows operating system. It was accepted with a three-month delay in March 2019.

2.2 Stand-alone nEYron

Once the POC deployment had been stabilized, the system was given a new name: nEYron. For two years, it was used by several EY employees on a single PC machine that hosted all system components. Meanwhile, nEYron acquired a new look, consistent with the style of other applications dedicated to the same customer. New functional features were developed to satisfy needs arising during the use of the application. An up-to-date list of functionalities is given in section 3.

2.3 Multi-user Solution

The final phase of deployment took place in 2021. The agreement stated that the application must adhere to EY security standards. The customer expected to receive the following items:

- system installation package;
- system installation instructions;
- system backup policy;
- user's guide;
- disaster recovery procedures.

The creation of the documentation was painless. However, adhering to the security standards was not (see 5.2). The process began in April 2021, and the certificate of final acceptance was signed in October 2021.

3 System Requirements for the Final Version

3.1 EY User Feedback

During the POC stage, EY employees developed a list of requirements that should be added to the system in the Final Version stage. The following three requirements were added after the POC stage: automatic deletion of documents from the user translation history after a specified time (for confidentiality reasons), document sharing between multiple users, and calculation of the approximate cost of translation of a document by a human translator before it is translated by a machine. Cost assessment was intended to help determine to what extent machine translation reduced translation costs over time, compared to human translation. It is based on the number of words included in the document. In addition to the updated list of requirements, EY employees in collaboration with the Poleng team created a mockup of the user interface that would correspond to the look and feel of the other internal EY systems. The user interface was further modified according to the EY guidelines during the development of the Final Version.

3.2 Final List of Requirements

The complete and up-to-date list of requirements consists of the following:

- user registration and login, including SSO (single sign-on) login, universal for all services accessible by EY employees;
- document import in `.txt`, `.docx`, `.pptx` and `.xlsx` formats;
- document editing in sentence-by-sentence mode;
- machine translation in an editing window;
- machine translation of entire documents;

- export of the translated document in a format compatible with the imported document;
- pre-translation of documents using translation memory fuzzy search matches;
- ability to proofread and approve translations of sentences;
- expanding translation memory with approved translations;
- transfer of document formatting (fonts, styling, text placement) between input and output document;
- archiving of translated documents per user;
- automatic deletion of documents from user translation history after a specified time;
- document sharing between multiple users;
- calculation of approximate cost of document translation by a human translator.

4 System Components

The architecture of the system consists of the following components:

- Machine Translation Service;
- Translation Memory;
- Web Application.

4.1 Machine Translation Service

Machine Translation Service provides translations of sentences in the English–Polish and Polish–English directions without human intervention. It is designed as a web service that is invoked by the web application to produce document translations. It is based on the Marian NMT framework (Junczys-Dowmunt et al., 2018). Internally, the web service forwards source sentences from HTTP requests to the Marian websocket server and returns the translations to the web application.

4.1.1 Customer Training Data

In-domain business documents translated by humans were delivered to Poleng in pairs: each document in Polish had its equivalent in English. The document format was either PDF or Microsoft Office (.docx, .doc, .pptx, .xlsx). We applied the following procedure to extract bilingual corpora from business documents:

1. Text extraction from business documents using the Apache Tika[1] toolkit.
2. Text segmentation into sentences using eserix[2] – an SRX rule-based sentence segmenter.
3. Text normalization, including punctuation, quoting and commas, using Moses (Koehn et al., 2007) scripts.
4. Alignment of a source text to a target text at the sentence level using the hunalign (Varga et al., 2007) sentence aligner.

This procedure initially allowed us to obtain nearly 70,000 in-domain sentence pairs.

4.1.2 Model Training

Model training consisted of two steps: training of general models on 10 million sentences derived from the OPUS corpora (Tiedemann, 2012), and use of the transfer learning paradigm to fine-tune the general models on the in-domain data. In this way, the system transfers the knowledge from the general model, significantly increasing the translation quality on the in-domain data (such a process has been described, for example, in Aji et al. (2020)). As the general model can be reused for future fine-tunings, this technique reduces the total time to solution by a significant margin.

Data preprocessing, in addition to using the Moses (Koehn et al., 2007) normalization scripts, included subword segmentation. We applied subword segmentation to the data using the SentencePiece (Kudo and Richardson, 2018) tool with the byte-pair encoding (BPE) (Sennrich et al., 2016) algorithm. The vocabulary consisted of 32,000 entries.

All NMT models were trained using the Marian NMT (Junczys-Dowmunt et al., 2018) framework on a single NVIDIA GTX 1080Ti GPU.

For the Proof-of-Concept stage, we trained models based on an RNN-based encoder–decoder architecture with the attention mechanism (Sennrich et al., 2017). We manually assessed translation quality, comparing the model trained only on openly available data with the model fine-tuned on in-domain data as described in section 4.1.1. The annotators evaluated the translations of a test set consisting of 488 sentences, and provided scores

[1] https://tika.apache.org
[2] https://github.com/emjotde/eserix

for accuracy and fluency by absolute grading on a scale from 0 to 5. The average scores obtained in all of these experiments are presented in Table 1. The most significant improvement in the fine-tuned version was achieved for translation accuracy in the Polish–English direction.

Direction	Data	Accuracy	Fluency
PL – EN	Open	3.47	3.61
EN – PL	Open	3.48	3.62
PL – EN	EY	**4.23**	**3.94**
EN – PL	EY	**3.90**	**3.74**

Table 1: Results of manual evaluation of preliminary experiments

The results of this manual assessment of the POC version were considered good enough to proceed to the next stage of deployment.

In the final deployment, the NMT model architecture was replaced by the base Transformer (Vaswani et al., 2017), which improved the quality of translation while reducing the time required to train the model. In addition, another 10,000 sentence pairs were derived from new documents provided by the customer. These additional sentences were used for training of the Transformer models.

The results of automatic evaluation based on the BLEU (Papineni et al., 2002) metric, calculated by the SacreBLEU (Post, 2018) tool with default settings, are presented in Table 2.

Direction	Data	Architecture	BLEU
PL – EN	Open	RNN	29.72
EN – PL	Open	RNN	26.36
PL – EN	EY	RNN	36.91
EN – PL	EY	RNN	32.99
PL – EN	Open	Transformer	31.13
EN – PL	Open	Transformer	28.34
PL – EN	EY*	Transformer	**39.92**
EN – PL	EY*	Transformer	**35.55**

Table 2: Results of automatic evaluation

4.2 Translation Memory

Translation Memory is a database of corresponding segments in both languages. The translation of a sentence is added to the memory upon approval by the system user. Search is carried out by an in-house solution: the Anubis system (Jaworski, 2013), which uses a suffix-array-based index for fuzzy matching. Anubis also features a unique algorithm for the detection and recombination of all sub-segment matches between a candidate sentence and an example from the Translation Memory.

Translation Memory serves two functions in the system: it is used during the translation process, and it also serves as a collection of training data for future fine-tuning of NMT models. During translation of a document, each sentence is first checked in the Translation Memory. If a match is found, the translation is returned as the result and the sentence is not translated by the NMT model.

4.3 Web Application

Web Application is the part of the system with which the user interacts. It consists of the following components:

- a server application, following the REST API design, written in the CakePHP framework;
- a user interface, written in the Vue.js framework;
- an SQL database.

All features included in the web application are listed in section 3.

Document translation process The main feature of the web application is the document translation process. It consists of the following steps:

1. User imports the document into System;
2. System extracts text from the document;
3. System segments text into sentences using SRX-based rules;
4. System checks the Translation Memory for the existence of each sentence;
5. System sets up batches of sentences whose translations have not been found in the Translation Memory;
6. Batches are sent to the Machine Translation Service;
7. System saves the translations in the database;
8. System prepares the document to be exported at user's request.

Translations found in the Translation Memory and translations produced by the Machine Translation Service are presented to the user in a single window. Once the document has been translated by the machine, the user can post-edit the text segment-by-segment. Each translated segment may be manually approved by the user for it to be stored in the Translation Memory.

Document reconstruction process The system is expected to transfer the document's styling and formatting from the source document to the translated document.

To this end, we make use of the Microsoft Office document structure: the document is unzipped into a set of XML files and the files are iterated in a search for text content. Each found text item is stored in a database and replaced in the XML file with a placeholder tag containing its identifier. When the translation of text items has been completed, the XML files are iterated again, and the placeholder tags are replaced by the translations. Finally, the XML files are zipped back into the Microsoft Office document package.

5 Deployment Challenges

5.1 Proof-of-Concept Deployment Challenges

During the POC stage, the entire system was installed on a single PC machine. The initial configuration of the machine and the installation of the system was carried out at Poleng's headquarters in Poznań, Poland. After the system had been installed, the machine was transported to EY's headquarters in Warsaw, Poland. For confidentiality reasons, the machine could not be connected to the Internet and any system updates had to be provided locally. Poleng prepared Docker[3] containers for each of the system components and transported them on a flash drive to the PC machine, when necessary. The use of Docker containers significantly simplified the process, as each deployment of a system update consisted of replacing the Docker container.

The only part of the system that could not be updated in this way was the NMT models. For security reasons, training of the model on customer data had to be performed on a PC machine at the EY headquarters. Therefore, the models were not part of the Machine Translation Services container. Instead, they were mounted as a volume in the container so that they could be easily replaced.

[3]https://www.docker.com

5.2 Security Check

For the deployment of the multi-user version in the EY infrastructure, each component of the system had to meet a list of security requirements. The necessary modifications to the Translation Memory and Machine Translation Service components were minor, as they involved only changes to the security of the Docker container (the main process running in the container could not run as a root user). The changes to Web Application were more significant, as this component is exposed to the user. The total number of security requirements that the web application had to meet was close to 70. Most of the security requirements (such as the setting of special headers in HTTP responses) were easy to satisfy. However, some security standards proved to be challenging. Among them were:

- replacement of the entire application logging module;
- implementation of the single sign-on (SSO) authentication procedure specific to the EY corporation;
- implementation of database encryption.

A thorough security review was performed by the EY Global technical team after the system had been deployed.

5.3 Installation in the Production Environment

Installation of the final version of the system in the production environment included the creation of the installation package and its deployment to the EY infrastructure. The installation package consisted of Docker containers with the system components. Each of the system components was deployed in Docker containers to enable system scalability in the future. The deployment process was executed through screen sharing. Poleng delivered the installation package to the EY technical team and guided them through the installation process.

6 Future Plans

Plans for the future include technical improvements to the existing solution, as well as the introduction of new features.

Small improvements may include replacing hunalign (Varga et al., 2007) with vecalign (Thompson and Koehn, 2019) in the bilingual corpus extraction process described in section 4.1. We expect that the translation quality of NMT models will improve as a result of better corpus alignment.

To further improve the quality of the NMT models, we intend to use existing monolingual customer documents. We plan to apply the back-translation (Edunov et al., 2018) technique iteratively (Hoang et al., 2018) to increase the quality of our models.

As new terminology emerges, the user expects MT systems to quickly adapt to them. In most cases, data that would cover the new terminology do not yet exist. To solve this problem, we intend to use techniques for forced terminology translation (Nowakowski and Jassem, 2021; Bergmanis and Pinnis, 2021) to ensure that specific terminology is translated according to the needs of the user. Additionally, providing a glossary with specific in-domain terminology would ensure the consistent translation of such terminology when different sentences are translated.

To date, we have relied on the BLEU (Papineni et al., 2002) metric for the evaluation of trained NMT models. To follow current state-of-the-art solutions in MT evaluation, we plan to use the MT Telescope (Rei et al., 2021) to evaluate our models with the COMET (Rei et al., 2020) metric and perform a fine-grained error analysis.

Business documents often have a complex layout structure, whereas current NMT models operate only on sentence-level textual semantics. We want to explore the idea of integrating NMT with Computer Vision to create an end-to-end model which would learn visual features, layout information and textual semantics to produce document-level translations better than the current state-of-the-art methods. Such a model would be able to simplify the process of text extraction, sentence segmentation and document reconstruction, as it would take all document information as an input. To this end, we plan to base our model on the TILT (Powalski et al., 2021) architecture. This was created for the Question Answering task, but we believe that it could be modified for NMT.

7 Conclusions

This paper has presented the deployment of an English–Polish translation system at the Polish branch of EY Global Limited. The system supports standard CAT and MT functionalities such as translation memory fuzzy search, document translation and post-editing, and meets less frequent expectations such as single sign-on login and calculation of the cost of human translation for a given document. The paper has presented the challenges that were faced during the deployment, particularly adherence to security expectations and installation in the production environment. Ultimately, the deployment took over three years. Meanwhile, new technologies have been developed in the field of Machine Translation. Once the security issues have been overcome, we hope to be able to update the system with emerging technologies, constantly improving its performance.

References

Aji, Alham Fikri, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. In neural machine translation, what does transfer learning transfer? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7701–7710, Online, July. Association for Computational Linguistics.

Bergmanis, Toms and Mārcis Pinnis. 2021. Facilitating terminology translation with target lemma annotations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3105–3111, Online, April. Association for Computational Linguistics.

Edunov, Sergey, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium, October-November. Association for Computational Linguistics.

Hoang, Vu Cong Duy, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia, July. Association for Computational Linguistics.

Jaworski, Rafał. 2013. Anubis – speeding up computer-aided translation. In *Computational Linguistics*, pages 263–280. Springer.

Junczys-Dowmunt, Marcin, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast

neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia, July. Association for Computational Linguistics.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Kudo, Taku and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November. Association for Computational Linguistics.

Nowakowski, Artur and Krzysztof Jassem. 2021. Neural machine translation with inflected lexicon. In *Proceedings of Machine Translation Summit XVIII: Research Track*, pages 282–292, Virtual, August. Association for Machine Translation in the Americas.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Post, Matt. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, October. Association for Computational Linguistics.

Powalski, Rafał, Łukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michał Pietruszka, and Gabriela Pałka. 2021. Going full-tilt boogie on document understanding with text-image-layout transformer. In Lladós, Josep, Daniel Lopresti, and Seiichi Uchida, editors, *Document Analysis and Recognition – ICDAR 2021*, pages 732–747, Cham. Springer International Publishing.

Rei, Ricardo, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online, November. Association for Computational Linguistics.

Rei, Ricardo, Craig Stewart, Ana C Farinha, and Alon Lavie. 2021. MT-Telescope: An interactive platform for contrastive evaluation of MT systems. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 73–80, Online, August. Association for Computational Linguistics.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August. Association for Computational Linguistics.

Sennrich, Rico, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nădejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain, April. Association for Computational Linguistics.

Thompson, Brian and Philipp Koehn. 2019. Vecalign: Improved sentence alignment in linear time and space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1342–1348, Hong Kong, China, November. Association for Computational Linguistics.

Tiedemann, Jörg. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey, May. European Language Resources Association (ELRA).

Varga, Dániel, Péter Halácsy, András Kornai, Viktor Nagy, László Németh, and Viktor Trón. 2007. Parallel corpora for medium density languages. *Amsterdam Studies In The Theory And History Of Linguistic Science Series 4*, 292:247.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Guyon, I., U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.