

Re-design of the Machine Translation Training Tool MT^3

Emiliano Cuenca
Paula Estrella
Laura Bruno

Faculty of Languages & FaMaF
University of Córdoba
Argentina

Emiliano.Cuenca@unc.edu.ar
Paula.Estrella@unc.edu.ar
lbruno@unc.edu.ar

Jonathan Mutal
Sabrina Girletti
Lise Volkart

Pierrette Bouillon

FTI/TIM, University of Geneva
Switzerland

Jonathan.Mutal@unige.ch
Sabrina.Girletti@unige.ch
Lise.Volkart@unige.ch
Pierrette.Bouillon@unige.ch

Abstract

We believe that machine translation (MT) must be introduced to translation students as part of their training in preparation for their professional life. In this paper, we present a new version of the tool called MT^3 , which builds and extends on a joint effort undertaken by the Faculty of Languages of the University of Córdoba and the Faculty of Translation and Interpreting of the University of Geneva to develop an open-source web platform to teach MT to translation students. We also report on a pilot experiment with the goal of testing the viability of using MT^3 in an MT course. The pilot allows us to identify areas for improvement and collect feedback from students on the tool's usability.

1 Introduction

Machine translation (MT) has made enormous progress over the past few years with the development of neural systems (Koehn and Knowles, 2017), and the translation industry has therefore been increasingly integrating it into daily workflow processes. This may have a direct impact on translators, who need to learn how to work with MT in a step called post-editing (O'Brien and Moorkens, 2014). It is therefore important to help translation students understand the underlying concepts of MT, including how MT systems are trained from bilingual corpora, what distinguishes systems from one another in terms of the algorithms they use, as well as the impact of their internal functioning on translation quality and, hence,

on the post-editing task. Even if the idea behind these systems sounds simple (producing the most likely translation), it can be challenging to explain to non-tech savvy students how these systems produce a final output. Currently, there are many open-source tools available to train statistical and neural MT models (the most widely used are Moses (Koehn et al., 2007) for statistical MT and TensorFlow¹ or OpenNMT (Klein et al., 2017) for neural MT) or commercial tools (such as KantanMT² or Microsoft Custom Translator³). However, it is difficult to integrate them in the classroom because open-source tools are mostly designed for IT professionals, who do not need a graphical user interface (GUI), as opposed to translation students; as for commercial tools, the costs are far too steep for some institutions.

The Faculty of Languages of the University of Córdoba (FL-UNC) and the Faculty of Translation and Interpreting of the University of Geneva (FTI-UniGe) have been collaborating since 2017 to design and prototype a tool that will support lecturers in the classroom.

In this paper, we present a new version of the tool called MT^3 , which builds and extends on a joint effort undertaken by both institutions to develop an open-source web platform to teach MT to non-technical students. We also report on a pilot experiment with the goal of testing the viability of using MT^3 for an MA course on MT. After the lecturer presented the topic to students, they had to carry out an exercise using the tool and then complete a questionnaire. The goal of the pilot was not only to test the tool and identify areas for improvements, but also to get subjective feedback from the

© 2020 The authors. This article is licensed under a Creative Commons 3.0 licence, no derivative works, attribution, CC-BY-ND.

¹<https://www.tensorflow.org/>

²<https://kantanmt.com/>

³<https://portal.customtranslator.azure.ai/>

students on its usability.

2 MT in the classroom

MT is one of the topics that should be introduced to translation students in preparation for their professional life, as lacking the necessary skill-set to take on post-editor roles or other technical roles, such as language engineer or MT engineer, could have a negative impact on their future employability, as many researchers and lecturers have observed (Lara, 2019; Koponen, 2015; Doherty and Kenny, 2014; Kenny and Doherty, 2014; Rico, 2017; Temizöz, 2016; Mellinger, 2017; Guerberof and Moorkens, 2019).

The EMT Expert Group (2017, p. 7) “recognises that the ability to interact with MT in the translation process is now an integral part of professional translation competence”. Furthermore, as part of the technological competence (p. 9), they highlight the inclusion of “basic knowledge of MT technologies and the ability to implement MT according to potential needs”. This reflects the current need of many translation and localization companies to hire specialists with a background in translation or linguistics who know how to train and assess the quality of MT engines. As Lara (2019), Kenny and Doherty (2014) and Doherty and Kenny (2014) explain, MT does not depend exclusively on informatics, but requires a close collaboration with language experts for the creation of corpora, labelling, quality assessment, terminology management, controlled-language definition and text preprocessing.

The importance of introducing these concepts as part of translator training lies in having a set of specific tools that can serve as a playground on which translation students can gain hands-on experience, while also learning about the concepts related to MT. As Pym (2013, p. 494) points out, “students should not learn just one tool step-by-step. They have to be left to their own devices, as much as possible, so they can experiment and become adept at picking up a new tool very quickly, relying on intuition, peer support, online help groups, online tutorials, instruction manuals, and occasionally a human instructor to hold their hand”.

However, a major roadblock when teaching technical content such as MT to non-technical audiences (like translation students) is a lack of suitable platforms that allow users to create MT engines without having to deal with low-level pro-

gramming languages, Unix consoles or command lines. This is the main motivation for prototyping *MT³*.

3 Support tools for teaching MT/PE

To our knowledge, there are only few active open-source platforms that can be used for teaching MT: Joey NMT (Kreutzer et al., 2019), MTradumàtica (Doğru et al., 2017) and Interactive Teaching Tool (ITT) (Khayrallah et al., 2019). However, they do not provide all the desired features, e.g. MTradumàtica and ITT do not provide neural MT (NMT) models, nor the possibility to visualize intermediate results, and Joey NMT does not have a GUI to train and test models. A valuable resource for practicing post-editing and evaluating MT systems is PET (Aziz et al., 2012), however, this tool does not offer an integrated MT module. Using several tools in an unintegrated way may present new challenges to students, such as compatibility (e.g. file formats) or confidentiality issues. It would therefore be desirable to work on the same platform with intranet restrictions.

The lack of free tools that meet our desiderata has fuelled the joint development of an open-source web platform called *Machine Translation Training Tool - MT³*. This platform aims at providing a playground for translation students and non-tech savvy users, helping them get hands-on experience with two main kinds of MT technology: statistical MT (SMT), which is less used in the market, but is helpful for pedagogical and comparative purposes, and neural MT, which is the current state-of-art and is more difficult to teach since it involves more complex algorithms and intermediate processing steps. The main goal of this tool is to make abstraction of the technical details by letting users focus on the important processing steps and helping them understand the internals by visualizing intermediate processing results.

Additionally, we consider *MT³* to be a teaching aid that may be useful for developing and testing a syllabus that focuses on MT topics, including practical activities that can be carried out using this platform; hence, it might be possible to measure the acquisition of competences through self-efficacy measures (Bandura, 1977, 2006; Compeau and Higgins, 1995). At the time of writing there is a similar initiative called *MultiTraiNMT - Machine Translation training for multilingual citi-*

zens.⁴

4 Previous versions of MT^3

The first version of this tool was created almost three years ago (Bouillon et al., 2017). In this initial stage, the tool worked only as a desktop application and consisted of a Python module that executed different scripts and processes in the background through a GUI. This GUI, shown in Figure 1, consisted of a series of tabs associated with the well-defined steps of creating a baseline engine with the Moses toolkit.⁵

Each tab consisted of a set of fields where the user inserted the arguments; a button that activated the task in question, for which the internal engine executed one or more sub-programs; and an area on which the output of these sub-programs was printed. With this version, it was possible to train and evaluate statistical models, as well as post-edit the raw output by using a simple table with source and raw MT.

The obvious drawbacks of this version are that the computer can become blocked due to the memory-intensive processes, and it is also highly dependent on electrical power (possibly for several hours or even days) to ensure that the computations that have been carried out are not lost.

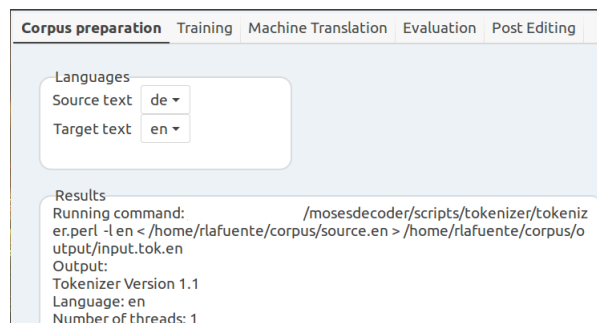


Figure 1: MT^3 version 1: screenshot of the Python-GTK GUI.

An improved version consisted of adapting the previous work to the web, while keeping the original distribution of tabs and a very similar look and feel. Besides overcoming the roadblocks of a desktop version, this version also incorporated the use of Docker (Merkel, 2014) to run all heavy processes on the server inside a Docker container; this

⁴Information available at <https://ec.europa.eu/programmes/erasmus-plus/projects/eplu-project-details/#project/2019-1-ES01-KA203-064245>

⁵These steps can be found at <http://www.statmt.org/moses/?n=Moses.Baseline>

allows for a quicker deployment and maintenance of the tool.

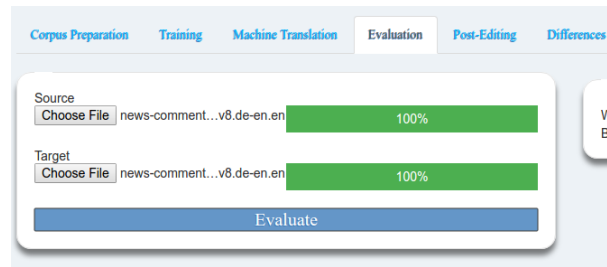


Figure 2: MT^3 version 2: screenshot of the Web GUI.

In addition to the above-mentioned functionalities, a new functionality was included to visualize the modifications made by the post-editor to the raw output, and some basic statistics, like the total amount of time spent on a segment and the number of edit operations, were also added.

Despite the improvements, some structural issues remained, since most of the work was done by students in Computer Science as part of their Masters' graduation projects. This version provided the starting point for the current version, which is presented in the next section.

5 Current version of MT^3

For the new version, the back-end was completely re-designed to provide a more reliable infrastructure that would be suitable for neural models. The GUI was also re-designed to include key functionalities that were missing in previous versions, such as user registration and authentication, or model sharing. Figure 3 shows the new architecture of the server with the major additions: the use of several (as opposed to only one) dedicated persistent Docker containers for special services (web API, database, task manager) and non-persistent containers for smaller tasks.

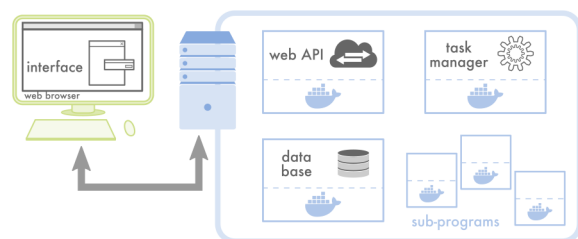


Figure 3: Proposed architecture for version 3 of MT^3 .

The following sections provide some details about the server and the client (user GUI).

5.1 Server-side architecture

The server uses a REST API, which is accessible through a network connection via the usual HTTP protocol. This type of architecture allows for a distributed computation of the tasks, so that the most demanding work is carried out on the server, while the client takes care of issues related to presentation and interaction with the user. This role assignment is particularly convenient as it allows the user to run the software on a machine with modest resources, without the need for any installation, since all operating systems include a web browser by default.

As for the building blocks of the server, we have based the new version on Docker⁶, an open-source Platform-as-a-Service (PaaS) that allows for the isolation of each module and its dependencies from the underlying operating system.

The server creates and destroys containers dynamically to execute tasks related to training statistical MT engines with Moses, neural engines with OpenNMT, and the application of standard evaluation metrics WER, TER (Snover et al., 2006) and BLEU (Papineni et al., 2002).

5.2 GUI functionalities

We have kept the main functionalities of the previous versions and added major ones, namely user management (register and authentication), persistence of user's data (models, translations and evaluations), neural engine training capability (based on OpenNMT's encoder-decoder model), visualization of intermediate results for statistical engines (an online visualization with no need to download) and engine sharing (for more ecological use of resources by avoiding training big engines with the same data multiple times).

Screenshots provided in Figures 4 to 7 show the current state of the tool. After logging, the menu on the left-hand side can be used to access the different modules (Figure 4). Data used in previous training sessions will appear in the right pane, as shown in Figure 5, where three statistical MT models appear: the first one is owned by our test user and the other two are shared by other users. Owned models can be deleted and any model can be explored, tested and used for translation. Figure 6 shows a shared engine being tested: the user is requesting the translation of *the cat is sleeping* and the result shows *le chat dort*. Note that these are

⁶<https://hub.docker.com/>

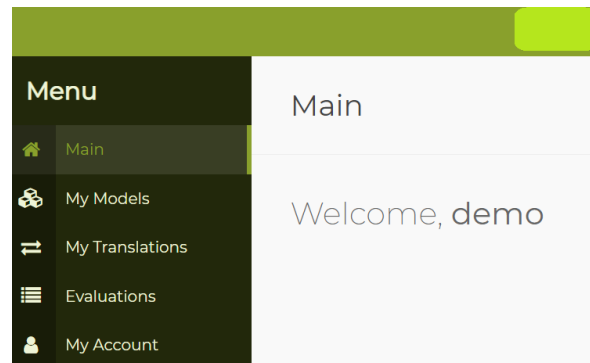


Figure 4: After successful login, the menu on the left shows the modules that can be accessed by the user.

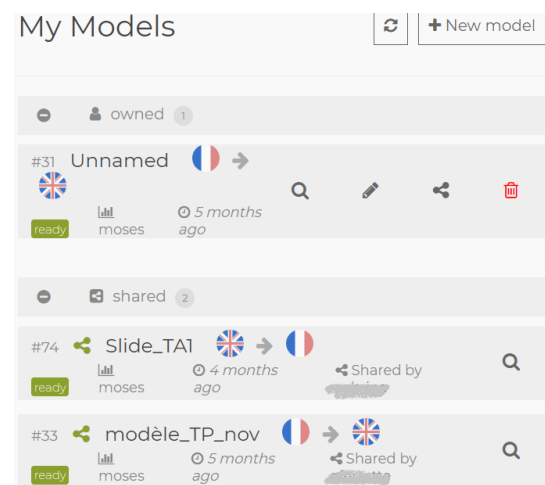


Figure 5: The interface shows the models owned and shared by other users

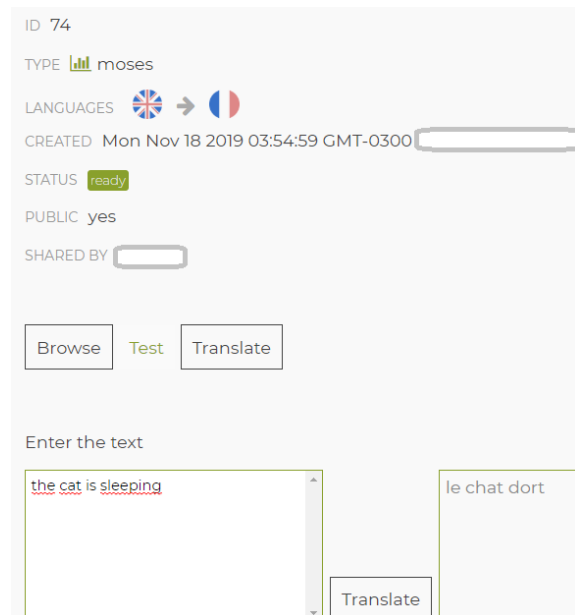


Figure 6: To test a model or run MT on a small number of strings, the “Test” option can be used; it is also possible to “Browse” the training files of the engine and to “Translate” files.

tiny models built for the purposes of a course on MT where the tool was tested, but the sharing option offers the lecturer the possibility of training a big, robust model, sharing it with students and letting them build smaller engines to compare their quality to the bigger one.

The functionalities to translate and evaluate MT output are very intuitive and follow standard practices; for example, Figure 7 shows the result of one of the evaluations.

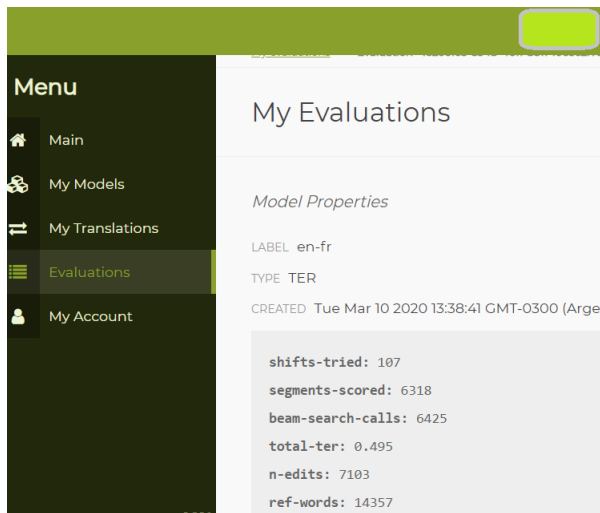


Figure 7: Automatic evaluations can be done using BLEU, TER or WER, and results are stored for future reference.

6 Pilot use in the classroom

In November 2019, a pilot test was carried out in the classroom as part of an MT course (“*Traduction automatique 1*”) at the FTI-UniGe. This experiment was aimed at evaluating the performance and usefulness of the tool for MT classes. Given that the tool is still under development, we decided to start testing parts separately and, at the time of the pilot, we focused on the functionalities related to statistical MT. As part of the development roadmap we will include a second pilot to test the NMT functionalities and add a post-editing tab.

For the first pilot, students were provided with a publicly available statistical model previously trained by the lecturer for the language direction French-to-English. They had to explore the files generated during the training phase and perform quick translations (tests) in order to answer the following questions:

1. What is the target language of the model?
2. Was the model trained with the phrase *Les avocats dorment*?

3. Is the word alignment correct for *aime/like*? Explain your answer.
4. Do the bigrams *avocados.* and *lawyers.* have the same logarithmic probability?
5. Has *J’aime* been properly segmented into two tokens during the tokenization phase?
6. Does the system have the translation of *j’* (in lower case) in the lexical translation model?
7. What is the probability of *hope - espère*?
8. If you write the phrase *J’espère les avocats* in the test interface, you will get the translation *I hope The lawyers*. Why is *The* capitalized?
9. What happens if we try to translate a sentence that contains a word that is not in the training corpus?
10. Is the model a bigram or a trigram one?

At the end of the class, students were invited to complete a survey on their experience with the tool.

The questionnaire addressed different aspects, such as the previous MT experience of each user. Given that the users were translation students, technical concepts related to the software were presented in an accessible way, striving to maintain a balance between simplicity and precision. The survey was distributed via Google Forms, and the most relevant questions were:

1. Did you have any previous knowledge of MT before taking this course?
2. Did you have previous knowledge of post-editing before taking this course?
3. What language pair do you work with?
4. Did you use MT^3 during this course?
5. How much do you agree with the following statement: “I have used MT^3 to practice and it has helped me to better understand the theoretical concepts presented in class”?
6. How much do you agree with the following statement: “The MT^3 interface is friendly and intuitive (i.e. users can perform assigned tasks without having to click here and there to find what they need)”?

7. How much do you agree with the following statement: “Considering that it is in development, MT^3 seems to be reliable enough to perform MT tasks (it is capable of performing the desired function without fail, or, in case there are failures, these are clearly reported to the user)”?”

The following are the results of the survey, obtained from the responses received. We are aware that the number of respondents is small (9 valid replies in total) as the survey was not mandatory. Nonetheless, it provides valuable initial feedback and the whole pilot served as an assessment of the development process, helping us to identify a roadmap for future use and development.

Of the total number of respondents, three had prior knowledge of MT and post-editing. All of them worked at least with the French-English language pair, while two of them also worked with the French-German language pair, two others with French-Spanish, one with French-Italian and one with French-Russian.

Replies to question 4 indicate that 4 of the respondents had only seen the teacher using the tool, without using it themselves, while another 4 had used it in class. The final respondent said that he had also used it from home.

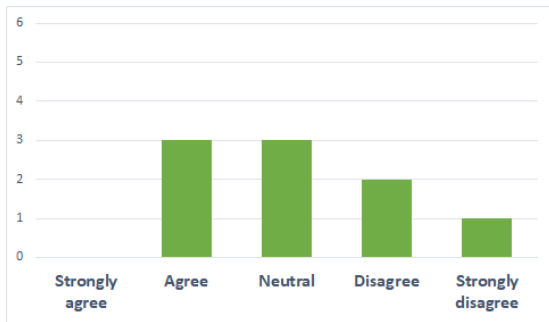


Figure 8: Results obtained for question 5, related to the usefulness of the tool as support for learning theoretical concepts.

As to whether the tool was useful in supporting the theoretical content, 3 agreed, while another 3 were neutral; 2 disagreed and 1 strongly disagreed. These results are shown in Figure 8.

As to whether the interface was friendly and intuitive, 4 respondents agreed, while 3 disagreed, and 2 were neutral. These results are shown in Figure 9. Regarding the reliability of the tool, 3 respondents considered the tool to be sufficiently reliable, as opposed to 1, who thought otherwise, while 5 respondents remained neutral, as shown in

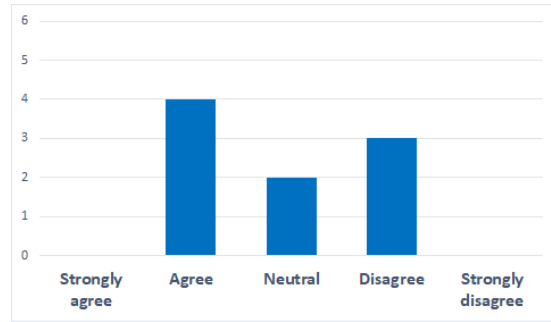


Figure 9: Results obtained for question 6, related to the user-friendliness of the GUI.

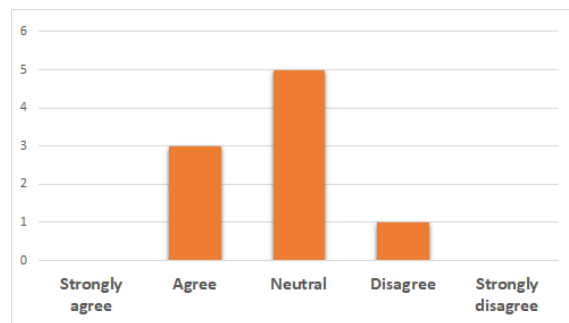


Figure 10: Results obtained for question 7, related to the reliability of the tool.

Figure 10.

Only one respondent reported having problems using the tool, particularly with the expiration of the session forcing him to log in again.

Finally, a student suggested highlighting some relevant files in the model, to avoid searching through all the files, which may generate some confusion.

7 Conclusions and future work

We have presented a new version of MT^3 , an open-source web platform intended for teaching MT to translation students and others interested in the topic, who have little or no technical skills. It allows the students to create statistical models using Moses and neural models using OpenNMT, which can be shared among other users. It also includes user management, such as authentication and a basic system of privileges to perform actions on certain resources. User activities, such as translations and evaluations, can be stored. In addition, it aims to show MT models as white boxes, enabling navigation and access to the files generated during the training process.

We have also reported on a pilot experiment based on a real-life MT course for translators,

given at the FTI-UniGe. Although the pilot gave us insights on the usefulness of the tool, the clear limitation was the small number of participants. The result of this experience was overall very encouraging, demonstrating that the tool can be installed and used to a certain extent as a support for practical activities in order to apply theoretical concepts.

The work carried out so far was not intended to achieve a final definitive version, but rather an intermediate step towards a more reliable and extensible platform that will serve as a starting point for future developments. These include the visualization of data created during the training of neural engines, the integration of a post-editing tab and a better logging mechanism to provide more specific error messages. As for the utilization of the tool, we are planning another pilot to further test it with students. Finally, we intend to make the platform available in the future to the translation teaching community.

Acknowledgements

This project was financed by SeedMoneyGrant 2019 as part of a joint research project between FL-UNC and FTI-UniGe.

References

- Wilker Aziz, Sheila Castilho, and Lucia Specia. Pet: a tool for post-editing and assessing machine translation. In *LREC*, pp. 3982–3987, 2012.
- Albert Bandura. Self-efficacy: toward a unifying theory of behavioral change. *Psychological review*, 84(2): 191, 1977.
- Albert Bandura. Guide for constructing self-efficacy scales. *Self-efficacy beliefs of adolescents*, 5(1): 307–337, 2006.
- Pierrette Bouillon, Sabrina Girletti, Paula Estrella, and Roxana Lafuente. MTTT – Machine Translation Training Tool: A tool to teach MT, evaluation and post-editing. In *The 20th Annual Conference of the European Association for Machine Translation (EAMT)*, p. 18, 2017.
- Deborah Compeau and Christopher Higgins. Computer self-efficacy: Development of a measure and initial test. *MIS quarterly*, pp. 189–211, 1995.
- Gökhan Dođru, Adrià Martín-Mor, and Sergio Ortiz-Rojas. MTradumàtica: Free statistical machine translation customisation for translators. In *20th Annual Conference of the European Association for Machine Translation*, 2017.
- Stephen Doherty and Dorothy Kenny. The design and evaluation of a statistical machine translation syllabus for translation students. *The Interpreter and Translator Trainer*, 8(2): 295–315, 2014.
- EMT Expert Group. European Master’s in Translation competence framework, 2017.
- Réka Eszenyi and Brigitta Dóczy. Rage against the machine—will post-editing assignments outnumber translations in the future? *Fit-For-Market Translator and Interpreter Training in a Digital Age*, pp. 119–133, 2020.
- Federico Gaspari, Hala Almaghout, and Stephen Doherty. A survey of machine translation competences: Insights for translation technology educators and practitioners. *Perspectives*, 23(3): 333–358, 2015.
- Ana Guerberof Arenas and Joss Moorkens. Machine translation and post-editing training as part of a master’s programme. *Jostrans: The Journal of Specialised Translation*, (31): 217–238, 2019.
- Dorothy Kenny and Stephen Doherty. Statistical machine translation in the translation curriculum: overcoming obstacles and empowering translators. *The Interpreter and translator trainer*, 8(2): 276–294, 2014.
- Huda Khayrallah, Rebecca Knowles, Kevin Duh, and Matt Post. An interactive teaching tool for introducing novices to machine translation. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education, SIGCSE ’19*. ACM, 2019.
- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*, 2017.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 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*, pp. 177–180, 2007.
- Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*, 2017.
- Maarit Koponen. How to teach machine translation post-editing? Experiences from a post-editing course. In *4th Workshop on Post-Editing Technology and Practice (WPTP4)*, pp. 2–15, 2015.
- Julia Kreutzer, Joost Bastings, and Stefan Riezler. Joey nmt: A minimalist nmt toolkit for novices. *arXiv preprint arXiv:1907.12484*, 2019.
- Cristina Plaza Lara. Swot analysis of the inclusion of machine translation and post-editing in the master’s degrees offered in the EMT network. *Journal of Specialised Translation*, (31): 260–280, 2019.

- Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. Visualizing and understanding neural models in nlp. *arXiv preprint arXiv:1506.01066*, 2015.
- Zachary C Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.
- Christopher D. Mellinger. Translators and machine translation: knowledge and skills gaps in translator pedagogy. *The Interpreter and Translator Trainer*, 11(4): 280–293, 2017.
- Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239): 2, 2014.
- Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73: 1–15, 2018.
- Sharon O’Brien and Joss Moorkens. Towards intelligent post-editing interfaces. In *Proceedings of FIT XXth World Congress*, Berlin, Germany, 2014.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics, 2002.
- Celia Rico Pérez. La formación de traductores en Traducción Automàtica. *Tradumàtica: traducció i tecnologies de la informació i la comunicació*, (15): 75–96, 2017.
- Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200, 2006.
- Özlem Temizöz. Postediting machine translation output: Subject-matter experts versus professional translators. *Perspectives*, 24(4): 646–665, 2016.