Using Neural Machine Translation Beyond Post-Editing

In the past, I've conducted a number of back-and-forth email conversations with experts on topics that are interesting and useful to me and, hopefully, to the community at large. The following conversation turned out to be very useful as well, but it was not conducted as straightforwardly as some of the others. Why? Well, it's because my discussion partner, Félix do Carmo, and I made certain assumptions as we communicated that the other either didn't understand right away or that were muddied by our own preconceived ideas. As a result, we went back and forth a number of times to amend our questions and answers. We soon realized that relying on assumptions about post-editing must also be a "problem" for others trying to have similar conversations. Indeed, it might be a symptom of many discussions, whether between the machine translation development community and translators, or even between translators with different specializations and language combinations, where the needs, tools, and language requirements demand different solutions.

Today I'm starting a conversation with Félix do Carmo, a translator and now also machine translation researcher, about better usability practices for the professional use of machine translation (MT). Félix, do you want to introduce yourself?

I'm the managing director of TIPS, Lda., a translation company specialized in Portuguese that I established with colleagues in 1994. I'm also a postdoctoral researcher at the ADAPT Centre at the Dublin City University, currently working as an EDGE Fellow in KAITER, a project which aims at studying and developing a tool that interactively learns and supports translators' editing work. My main interests are in the application of machine translation processes as supports to translators, covering areas like post-editing, human factors, machine learning, and translation tools.

I started teaching translation technology to university students and teachers in 1998. In 2010, I took the opportunity to work on a PhD, which allowed me to learn and collaborate with computer scientists and get to know the insides of MT. My project, which I finished in 2017, focused on studying how to describe and support post-editing. I then received a fellowship to the ADAPT Centre, which has allowed me to work as a researcher with people like Joss Moorkens, Dorothy Kenny, and Andy Way and to try to influence MT researchers to develop tools for translators, rather than autonomous devices. So, although I'm not producing translated and revised words, I see myself as a translator, playing different roles in the world of translation. I also take advantage of every opportunity to learn as much as possible about my profession.

To be in the driver's seat, translators will need to have a clear right to manage the data they produce.

Like you, I've also been interested in working with MT, not so much from the angle of traditional post-editing, but more in terms of using MT suggestions as one of a number of data sources to help translators in the translation process. I've been particularly eager to find good ways to use translation environment tools to semi-automatically use partial data from MT segments. That certainly seems to be a good method when working with statistical MT. I wonder what difference it makes that we now (typically) use neural machine translation (NMT). Are the results of NMT usable in the same way as the results of statistical MT?

That's an interesting question. To answer it, we probably need to start with some technical information about the different systems. In statistical machine translation (SMT), the decoder that "translates" is essentially a search algorithm. For each word and group of words in the new source sentence, the search algorithm consults the phrase table, which contains aligned words and groups of words from the training data, and extracts the best equivalent. So, the approach is paradigmatic: each source word creates a slot, which may be filled in by any word in the phrase table. The search algorithm looks for the best fit, as if it was looking for LEGO pieces, slotted into position in a vertical, top-down movement. This means that the resulting sentences are sometimes awkward, with syntactical errors and elements that don't go well together.

The decoders in NMT work differently. The decoder doesn't search for LEGO pieces from tables of aligned phrases. Instead, it first uses neural networks to learn and then identify the best sequences to translate full sentences. This is done from the mathematical representations of the sentences it learns from large amounts of parallel data. This mathematical data is only converted into words in the last stage of composing the translation. NMT tries to construct a sequence horizontally (linearly), not from the top down, but beginning to end, with each sequence of previous words determining the next word. So, it's as if the system works syntagmatically. First, it learns the design of the puzzle and then it learns which pieces form that design. That focus on the sequence, the syntagmatic view of language, is what makes NMT more fluent than SMT, since the connection between the elements that compose a target sentence are more tightly knit together.

So, when you and I think of MT output being disassembled into pieces that may be fed separately to a translator, we're thinking in terms of the SMT models, but this doesn't describe what happens in typical NMT models. NMT is

This column has two goals: to inform the community about technological advances and encourage the use and appreciation of technology among translation professionals.

not conceived to output partial data, but whole sequences.

But SMT was not conceived to be used that way either, but just happened to be generated that way. Wouldn't it make sense to say that a translation suggestion that comes from a neural engine also has valuable parts, regardless of whether the entire sentence sounds more fluent as a whole? And also independent of whether the suggestion was put in there as parts (as in SMT) or in the sequential manner you're describing for NMT? If that's so, then I don't completely understand why an automated fragment search doesn't make sense when working with NMT. But I'm very interested in what we can do with the MT suggestions once the (noninteractive) MT engine has "done its job" and presented the suggestion within the translation environment tool. Technologically speaking, now it's the task of the translation environment tool to present the usable parts of the suggestion. Speaking from a workflow perspective, this typically means that it's the translator's keystrokes that enable the tool to present suitable fragments.

You're right. If we start from the point of already having full suggestions and want to know how to extract information from them, then we shouldn't be discussing NMT and whether it fundamentally affects this process of choosing the best solutions. Like you say, it's no longer the task of the MT engine but that of the translation environment tool to present the words you want to use from the full suggestions it receives.

Again, this is a search problem, and there are many approaches for these complex problems. The sheer nature of linguistic data, which is so variable, makes searching linguistic items an even harder problem than usual. You suggested that typed keystrokes should bring up the correct suggestions from the different sources you have, but can you be sure the full suggestions from MT engines contain the words you want to write? For example, you may have several synonyms in two or three different suggestions, but not the one you're looking for. So, it's probably not enough for the algorithm to do a simple search in these suggestions. Instead, it will need to look in other sources (perhaps monolingual data) for the word you're typing. But under which conditions or rules should this search be done for it to be effective and efficient?

The main thing about tools that are adapted to specialized translators is that they should work in the background to feed the best suggestions possible to the translator, but the decision making needs to be done by the translator.

I agree that the current search mechanisms that are based on keystrokes are not advanced. There are no fuzzy features, or there is certainly no linguistically-driven search for synonyms or the like, but maybe that's not even what's needed. After all, the translator may not want to see a fuzzy match or a synonym if they've already decided to go with a certain term. What I take from this, though, is that there is no real difference in "harvesting" fragments from previously generated MT suggestions, regardless of whether they come from SMT or NMT. What other developments or perhaps under-researched areas are you looking at that would make NMT useful beyond "just" post-editing it?

Let's think about the current scenario in the translator's desktop, in which, as you say, MT suggestions can be used "as one of a number of data sources to help translators in the translation process." Although new sources of data bring new solutions, they also bring new problems. We could say that the impact of NMT in the translator's desktop is still globally under-researched. Let me discuss a few examples of issues that are not currently being researched enough.

NMT still requires very large amounts of training data, resorting to more data than translation memories usually hold. This means that NMT will always present hypotheses that will create new conflicts with a translator's local resources. Although research says that NMT produces "better output," this definition of quality is usually measured in isolated and simulated scenarios. We need different evaluation factors and metrics

to understand how useful "better output" actually is in real scenarios.

For us to discuss how we can move "beyond post-editing" and help translators develop new ways of working, we need to talk about the translation process itself. I believe there's still too much fog created by the introduction of the term "post-editing" in the industry, and we need to take a step back and try to get a clear view of what we call translation and what we call post-editing. Let me try to briefly express my view on this.

If your system feeds fragments of suggestions to translators so they can write the translation, then they are actually translating, not post-editing. Translators have to generate the translation in their mind before choosing to accept or change each word or phrase that is being presented dynamically to them. That's why we talk about a high cognitive load in this process, because the translator's thoughts are constantly being interrupted by the support system. Most of these systems are known as "interactive machine translation," but I would call the process "interactive human translation," because the resulting translation comes from that mental process. There isn't enough research on these cognitive loads and the effects of such things as increased productivity in a regular work life.

Post-editing, on the other hand, essentially involves editing. This is only possible when translators are presented with a full suggestion by the MT system that's good enough for them to read. Instead of thinking about a full translation alternative, translators are able to identify parts of the suggestion that require editing. In a post-editing project, translators edit some sentences, but they may also need to translate quite a few. So, post-editing involves both editing and translating. The threshold from which a translator is no longer editing but is actually translating is another under-researched area in which I'm interested.

But let's not fool ourselves into thinking that when we talk about editing, we're talking about a simpler task that's easier to learn and automate. If we go back to the paradigmatic and syntagmatic approaches (one approach identifying slots and filling them in, and the other approach more concerned with the relations in a sequence), we find that even editing involves those two dimensions in a decision process that's very difficult to predict. Editing may be broken down into four actions: deleting, inserting, replacing, and moving words. Only replacing is "simply" paradigmatic: you identify a slot that's occupied by the wrong word and replace it. Moving a word is a good example of a syntagmatic action because you mess with the structure your MT suggestion constructed. But estimating these actions isn't easy. It's been demonstrated that estimating all options of new positions of an element in a sequence is one of the hardest mathematical problems you can ask a computer to do. Again, more research is needed into the patterns of editing and how to create assistants that support these processes.

As our industry matures, we should identify the value of each node in the supply chain and adapt technology and management of resources to each of those nodes.

That's really interesting, but really theoretical. What's being done in academia with NMT in a more practical manner to move beyond "post-editing," as vague as that term might be?

I would say that current research is still very much focused on using and applying NMT to produce better output to feed to traditional translation tools. Here we should mention four areas of current research that will affect the way NMT output will be presented to translators: INMT, AMT, APE, and QE.

1. Interactive Neural Machine Translation (INMT) is dedicated to developing ways to incrementally feed output to translators from neural networks trained on parallel corpora. These systems model the translation work as described above. The translator generates the translation, starts writing, and the NMT system suggests the next fragment. If all goes well, the translation is created faster than if the translator didn't have this "voice over the shoulder." For these systems to be accepted and become regular tools translators use, they need to feed suggestions that are adjusted to each context. Since INMT outputs words that are constrained on the words already written, there's the expectation that the suggestions presented by these systems will be better than those possible with SMT engines. However, this is still an area that raises more questions than answers. For example, can you constrain the output not just on the previous target words, but also on a list of validated terminology, and control how accurate the whole process is?

2. Adaptive Machine Translation (AMT) has been proposed as a term to describe systems that learn the specific traits of each translator's work and adapt suggestions to those traits. It's not yet clear how this will be done, which traits these are (some refer to this as "style," which is one of the vaguest terms one can use), and how effective this actually is.

3. Automatic Post-Editing (APE) is another complementary area that's being researched. The name may sound like

another way to replace translators, not only in the translation stage but also in the editing and revision stages. Actually, I would say that APE is just another way to improve the output. It has been shown that applying NMT technology to APE improves the output of MT systems. However, again, despite the improvement in the output, this doesn't change the nature of the translating/editing work that's required or the fact that this work still requires professional translators.

4. Quality Estimation (QE) tries to provide some indication of the segments that may not require much editing, as well as those that may require extensive translation work. QE may also serve to highlight words that are probably wrong in a translation suggestion. This is complementary information that may help in the translation decision process. The use of NMT methods for QE has also enhanced the capacities of QE methods.

So, INMT, AMT, APE, and QE complement each other in helping the translator. They provide translators with better suggestions (either interactive/dynamic segments for them to use to build a translation, or else better full sentences for them to edit). They also help filter out bad suggestions and guide the translator's attention to those areas that may require more work.

To describe how to leverage this technology to provide translators with more than just better output for them to edit, discussions have focused around terms like "augmented translation" or "knowledge-assisted translation." Such discussions actually began a few years ago when we started talking about the next generation of translation tools. Apart from the integration of some of the concepts above, like INMT in Lilt or QE in Memsource, most of these ideas still haven't become a reality in the daily lives of most translators.

Academia and the industry tend to spend more time discussing the names for technology than on making the revolution happen. One of the most recent signs of that is the suggestion to stop talking about NMT (because it's said that it's now officially the same as MT), and to talk instead about artificial intelligence (AI). But all these new terms simply express the challenge to combine not just the plethora of sources we mentioned earlier, but also the plethora of technological approaches into the same tools.

There's still too much fog created by the introduction of the term "post-editing" in the industry, and we need to take a step back and try to get a clear view of what we call translation and what we call post-editing.

I really like the suggestion about talking about AI instead of NMT. It's also interesting to see that some of these areas of research have not only found their way into the tools you mention, but also tools such SDL, Intento, and ModernMT. As a final question, I would like to ask you something practical. The typical translator doesn't have access to customized MT engines (with the possible exceptions of the adaptive engines mentioned above, or if the client provides access to a customized MT). If translators choose to use an MT engine, they will end up using engines like Google, Microsoft, or DeepL. How can one of these engines—or indeed several at the same time—be used more productively or creatively than just having translators essentially respond to the suggestions these engines make? How can translators be in the "driver's seat" when using these resources?

For me, the next technological step will be personalization. Actually, it's not such a ground-breaking proposal, since this is another buzzword that has been hanging around for a while.

As our industry matures, we should identify the value of each node in the supply chain and adapt technology and management of resources to each of those nodes. Corporations will go on managing big data, but they will suffer from the anonymity and genericity of that data. Language services companies will need to manage their client's data judiciously, and freelancers will need tools that help them manage their own data locally.

So, to be in the driver's seat, translators will need to have a clear right to manage the data they produce. They will also need to be allowed to keep personal translation memories of all translations they do, as well as have more access to other translators' and companies' resources and to an increasing number of tools and technology. Translators will need to know their work better. They will need tools that record and give them better insight into what they've done in previous projects, whether these are individual projects or collaborative ones.

The translation tool will receive input from MT engines, translation memories from personal, client, or collaborative projects, terminology databases, previous answers to queries, online discussions on translation suggestions, and many other resources. As such, translators will need various tools (see below).

The main thing about tools that are adapted to specialized translators is that they should work in the background to feed the best suggestions possible for the text. Ultimately, though, the decision making will still need to be done by the translator.

As for the details of how to use these technologies productively and creatively, instead of just responding to suggestions, let's think about a futuristic scenario in which translators work in a mode simply called "interactive translation." This scenario would integrate MT and translation memory, different text resources and online features, and support both translating and editing work. It would also support both "interactive" and "pre-translation" translators: those who prefer to type over some text, and those who prefer to write from scratch.

In interactive translation, everything would come down to the challenges of building a good interaction with translators, and this means having an interface that adapts dynamically to their needs. Let me describe some features I envisage for this future adaptive tool.

The interface should be very clean and uncluttered at the beginning of the translation process, helping translators read the text that needs translating, perhaps even presenting them with an automatic summary. This interface might also show translators other projects in their pool of resources that may be associated with that text, as well as the main terms and segments that might prove problematic throughout the translation. At this initial stage, the tool will provide very detailed statistics that estimate the amount of effort necessary and the quality of the MT output. It could also provide other details that might be useful for more advanced users, such as the capability to extract rules from style guides and client instructions to help automate the review process.

Translators will be able to approach the translation in many different ways (e.g., working from the first segment to the last, or starting with those that are problematic). In the background, the tool will select the best resources for each segment—either a translation memory, a solution provided by an MT engine, or a composition from fuzzy matches, terminology, and any other resources.

When translators start work on a text, they will see the best suggestion the tool provides for each segment. If a suggestion is a perfect fit, they will be able to validate it. If they want to know more about a suggestion, translators will have a simple way to dig deeper and find where it comes from, how reliable it is, or if there are other alternatives from more preferable sources. Translators will also have the option to act on these suggestions one by one or to aggregate them (e.g., dealing with all full matches from a reliable translation memory at once). But if the suggestions provided need editing, translators will have several forms of support that I'll describe in a bit more detail below.

Although research says that neural machine translation produces "better output," this definition of quality is usually measured in isolated and simulated scenarios.

The suggestions from the tool will always be presented in full, but translators will be able to manipulate them (e.g., by moving things around, deleting words, or inserting new ones). When they select a word to apply any of these actions, the tool will adapt and show different supports. For example, when translators decide to replace words without moving them, the system should be ready to present alternatives for that position (e.g., perhaps simply a change in the form of that word). When translators move words around, the system should be able to suggest changes that depend on the new position of those words. The suggestions provided will not be the same for each translator or for each project. So, it will be fundamental that the tool learns from the translator's behavior (e.g., to predict regular edits and to save and reuse them in similar contexts in other projects).

There are other activities translators do that may be supported by these new tools, such as web searching or making annotations and queries. The knowledge behind decisions supported by these resources is not currently integrated into translation tools, and it would be great to have this closer at hand.

When translators stop work, the tool would be able to provide them with statistics on how far they are in terms of the whole project, or other assignments they are currently working on, and how the project is going in terms of final checks. Before submitting the finished translation to clients, the tool would do a QA check and reuse the records of the decisions made to guide translators when revising the project. For example, the tool might help translators prepare a report for the reviser that includes the most troublesome passages or a list of the sources that were used for new terminology.

The main thing about tools that are adapted to specialized translators is that they should work in the background to feed the best suggestions possible for the text. Ultimately, though, the decision making will still need to be done by the translator.

We could go on dreaming of the details of such tools, but our dreams as translators are not the same for everyone. For example, you and I realized during our conversation that you dream of tools that are not so focused on editing as the tools I think about. The tools you envision do not play such an intervening role, but rely on the translator generating the translation.

But the main idea I take from our conversation is how we moved from the impact of existing technology to a discussion on how we use it. For me, this is the right way to discuss technology. Conversations should not be dominated by a fear of how MT or any other technology determines our work methods or even the definition of our tasks. Instead, we should be talking about the type of research that focuses on the technology we need. There's still a lot of research to be done on individual working methods and how these change according to the project, motivation, or even mood. It was great to see how you and I share the excitement to think in terms of the future, and to try to imagine how current and new generations of translators will use smart tools that adapt to them.

Jost Zetzsche is chair of ATA's Translation and Interpreting Resources Committee. He is the author of *Translation Matters*, a collection of 81 essays about translators and translation technology. Contact: jzetzsche@internationalwriters.com.