

Post-editor shortage and MT

Lori Thicke

Machine translation (MT) technology is capable of revolutionizing our industry by delivering increased productivity, faster throughput, reduced costs, and improved quality and consistency. However, today's shortage of post-editors runs the risk of slowing MT progress. The question is, what factors are to blame and what can we do about them?

Back when my company LexWorks (formerly Lexcelera) started deploying MT, the general perception was that MT was five years away from being good enough – and always would be. What has happened since then? MT has improved. Progress has been made in a number of areas, particularly with the hybrid engines that combine the best of rule-based machine translation (RBMT) with statistical machine translation (SMT) techniques. But these changes have been incremental rather than groundbreaking. MT alone is still not capable of delivering fully automatic human quality translations.

So why, then, is MT on the roadmap of almost every savvy translation buyer today? If MT's quality hasn't improved as much as we had hoped it would, there has been at least one significant change in the MT landscape: our expectations. We have stopped expecting MT to be perfect. Instead, we have realized that there is a place for imperfect MT, and when it needs to be perfect, a strong business case can be made for human rework.

The latter means that if MT is still not perfectly fine out of the box, it's not a serious problem. It's still possible for a linguist to improve the raw output more quickly than translating the same text from scratch. When a linguist, known as a post-editor in this context, corrects raw MT output, productivity

gains can run from 20% to 50%. Considering that the results can be virtually indistinguishable from a traditional translation, using MT sounds like a no-brainer.

There are some caveats, of course. The MT engine must be the right engine. The content must be the right content. And most importantly, the engine must be properly trained for the content, language pair, domain and even, at a more granular level, product line.

Complex technology though it may be, this is not what is limiting MT to the early adopters. Rather, it's the lack of human resources. The issue that you hear echoed throughout our industry is that there are simply not enough post-editors willing to work with MT.

If my company has managed to retain a large pool of post-editors, it may be because we understand one basic truism: post-editors really hate poor MT output. They hate it so much that last year, post-editor anger over being asked to mop up bad MT was the most hotly debated topic on LinkedIn's Automated Language Translation Group. And who can blame them?

I believe the main reason for the shortage of post-editors is that too many companies disrespect their time and abilities by churning out poor quality MT and expecting them to work miracles, and to give a sizable discount while they are at it.

In their haste to respond to cost-cutting demands, whether from customers (as is the case with language service providers) or from upper management (as may happen within the enterprise), inexperienced MT practitioners may be guilty of pressing into service the first engine they think will do the job. Lacking the expertise to know which engine to use and how to train it so it performs well, these inexperienced companies are inadvertently burning out the post-editors who are left with the clean-up jobs. This inexperience also means that opportunities to train post-editors properly may be missed, and that post-editors may be regarded simply as end-of-chain workers rather than as key contributors to the process of improving MT.

It is incumbent on companies to make sure that post-editing isn't thankless drudgery with errors that won't stay fixed and lower pay for more work. Hence, here are some ways to keep translators happy about post-editing MT.



Lori Thicke is cofounder and general manager of LexWorks, cofounder of Translators without Borders and a member of the MultiLingual editorial board.

Choose the right engine

For optimal quality, the first and most important question is not which engine to choose. The debate around whether to use a rule-based or statistical engine treats MT as a tool. MT is far from a single tool – it is a process. Part of that process is determining the best tool to use with a given type of content, language pair, workflow and quality requirement. You also have to consider the available resources, including human, technical and data.

There are cases when a rules-based approach works best, cases where a statistical approach works best, and cases where a hybrid of the two works best. The trick is knowing which engine to use in which situation. Because the MT process impacts the quality that post-editors are given to work with, it makes sense to do some testing to begin with. There are, however, some basic rules of thumb to guide you.

Language is one of the most important determinants of engine performance. Some languages, such as French and

Spanish, tend to work well in most engines. That is to say, these languages may be handled equally well by engines such as Moses, n.Fluent, Safaba, Asia Online, Language Weaver, Bing and Google on the SMT side, and by engines such as SYSTRAN, Reverso, PROMT and Apertium on the RBMT side. In our experience, languages such as Japanese and German perform best with an RBMT approach such as SYSTRAN's. But once you leave the dominant languages, by far the greatest number of language pairs do not exist in an off-the-shelf RBMT engine. In this case the optimal choice is easy: training an SMT engine from scratch.

On the other hand, if you do not have enough data – we're talking millions of segments of in-domain bilingual and monolingual segments – you may not have enough corpora to train an SMT engine, so a pretrained engine like Microsoft Translator or Bing with your own domain adaptation might be the best choice.

There are also other factors to consider. If the terminology is fixed in a narrow domain such as automotive or software documentation, RBMT or a hybrid is generally the best choice. This is because the rules component protects terminology better. Wild West content where the terminology runs all over the map and would be impossible to train for, such as patents, works better with SMT. However, if there are metadata tags, you should be aware that SMT doesn't preserve tags well, so RBMT or hybrid technology will save you some headaches. The source of the content is also important – SMT is better suited to user generated content such as forums, whereas RBMT is better suited to documentation that needs to be post-edited to human quality.

If it seems a bit complicated to know what engine to use, one best practice we use is testing. Being engine-agnostic means that before starting any project we fully test our assumptions by running the content through an SMT engine, an RBMT

YOUR COMPLETELY INDEPENDENT TRANSLATION MANAGEMENT SOLUTION

"Now our translator can focus on localization and not recreating the document. The efficiencies we've gained have been as dramatic as moving from a typewriter to using Microsoft® Word."

Daniel Ng | **GPRO Technologies**



- Side-by-Side Translation Editor
- Translation Memory Suggestions
- Termbase Management
- Advanced Project Reporting and Statistics
- Unicode Support for Western and Eastern European, Double-byte Asian Languages
- Import/Export Multiple File Types, Including:
 - Adobe® FrameMaker® and InDesign®
 - Microsoft® Word and PowerPoint®
 - HTM/HTML, XML, XLIFF and More

Sales@MadCapSoftware.com | +1 858 320 0387 | MadCapSoftware.com

Copyright © 2012, MadCap Software, Inc., and its licensors. All rights reserved. MadCap Software, the MadCap Software logo, and MadCap Lingo are trademarks or registered trademarks of MadCap Software, Inc., in the United States and/or other countries. Other marks are the properties of their respective owners.



engine and a hybrid. The best quality output is the one we put into the post-editing process. After all, we owe it to the people we entrust with post-editing to make sure they have the best quality to work on.

Train the engine right

An MT engine that is not properly trained on your material and your terminology, whether rules-based or statistical, is going to waste post-editors' time as they correct errors that ought not to be there in the first place. It should go without saying that an engine that is inexpertly trained, or based on poor-quality data, will not encourage your post-editors to take on another project.

When you train a rules-based engine, it already "knows" the language, so what you are training it on is terminology. With RBMT, one of the advantages is that you can be sure that it will use the terminology it was trained on. This is why it is easier to post-edit RBMT or hybrid output. Today's SMT systems are still hampered by a lack of predictability, which means that translators waste a lot of time verifying terminology that already ought to be automatically verified. With an RBMT system that is correctly trained, the terminology has

been hard-coded, so post-editors know it's the right terminology. They can see that a term came directly from the client, domain and even product-level glossaries. SMT post-editors, on the other hand, complain what a time sink it is when they have to constantly verify that the terms employed are the right ones.

Let post-editors improve the engine, and quickly

Post-editing may seem at first glance to be less fulfilling work than translating, because basically it means correcting some pretty dumb errors. To combat this, a best practice is to involve post-editors in the challenge of coming up with ways to not only improve the text they are working on, but to improve the system itself. While most post-editors aren't engineers, as qualified linguists they can certainly, if asked, tell you which improvements are one-offs and which ones need to be made in the engine to improve quality. Asking your post-editors for feedback on improving your linguistic systems engages them more fully, increases their satisfaction and helps you build better MT engines.

It can be frustrating for post-editors to be faced again and again with the same issue they've already fixed. When post-editing SMT, that next training cycle may be six months or a year away because you usually want a fair bit of new data accumulated before you begin the process of retraining. In this case, the post-editors are not empowered to make lasting changes and it typically takes until the next training cycle to see any progress at all.

The picture is different with RBMT in the sense that as soon as errors are identified, they can be corrected in the engine. Directly. Changes take just a few minutes and engine performance rapidly climbs. An area of improvement for SMT engines – and some are already doing this – is to make those training cycles shorter so post-editors can see that their corrections have been taken into account, and also get the benefit of everyone else's corrections on the same text. Implementing corrections quickly is key to improving MT quality and reducing post-editor frustration.

Train your post-editors well

Post-editors do not necessarily need to undergo university training. On the

other hand, setting them to work with no training at all is a mistake. Post-editing is not the same as editing, and MT output is different from translation memory fuzzy matches in some fundamental ways.

Let your post-editors know what they should expect from your MT output and how the errors may be different from a traditional translation. Depending on the engine you are using, tell them what they need to watch out for. If the output is statistical, they should know not to be swayed by those fluid sentences and to be extra vigilant against a missed word that changes the meaning entirely (such as not). Also, terminology will need to be verified.

With RBMT, the post-editors will ideally work in an environment where they can see which terms came directly from the glossary, so they will not need to check them. But they will need to spend extra time on sentence structure because RBMT is generally more awkward.

Also, post-editors will need to know what level of post-editing you expect. That is, what level of quality you need and are willing to pay for. Light post-editing aims just to make the text understandable, while full post-editing results in quality that is indistinguishable from human quality. They'll need to know which level of post-editing you are asking for. We find it also helps to set expectations for how quickly they should be working. For light post-editing it could be 20,000 words per day, while human quality usually is connected with a speed of 5,000 to 8,000 words per day.

Post-editors are essential to obtaining quality MT. Luckily, it is likely that the pool of available post-editors will increase once we respect their time by choosing the right engine, training it properly and updating it often; respect their talents by involving them in the process; and set their (and our) expectations correctly.

You'll know when your MT process is working. It's working when translators want to do post-editing work instead of complaining bitterly about it. Even better, you'll know it's working when they write you testimonial letters about how surprised they are by the quality and how fast it enabled them to work. By the way, that really happened. **M**



The World at Your Fingertips with euroscript's GCM Solution!

Customers buy global products and services but want local information. They need a solution to help them launch their services and products in all targeted markets simultaneously, faster and more effectively.

Discover euroscript's Global Content Management solution: an innovative approach to localization and translation services.

euroscript

More than 18 offices worldwide
contact@euroscript.com
www.euroscript.com