JoSTrans
The Journal of Specialised Translation

www.jostrans.org · ISSN: 1740-367X

Thicke, L. (2013). The industrial process for quality machine translation. *The Journal of Specialised Translation, 19*, 8-18. https://doi.org/10.26034/cm.jostrans.2013.419

The industrial process for quality machine translation

Lori Thicke, LexWorks (Eurotexte Group) | Translators without Borders

ABSTRACT

Machine translation (MT) is not a tool. Machine translation is an industrial process. Selecting the right MT engine – rules-based (RBMT), statistical (SMT) or hybrid – is just one part of a process that, if correctly managed, is capable of lowering translation costs, increasing productivity and even improving quality and consistency. To reach this goal, the MT process must pass through consultation, content, customisation, piloting, processing, post-editing, metrics and maintenance. This article looks at the first three stages in the MT process – consultation, content and customisation – and how the virtuous circle of post-editing feedback supports quality MT output. This is important because if the MT process is badly managed, it is inevitably the post-editor who pays the price. A system that is based on post-editors cleaning up bad MT is just not sustainable. With quality as the goal, the question is not so much what engine to choose but what engine and what process will give the best results. To determine what activities are most effective in achieving MT quality, a LexWorks study showed that a well-trained engine with an ongoing cycle of improvements from post-editing feedback is essential for MT quality.

KEYWORDS

Machine translation, MT quality, productivity, MT process, post-editing, engine training.

Lexcelera, the company I co-founded in 1986, has been deploying Machine Translation (MT) systems for customer content since 2007, localising on a variety of engines in over a dozen languages. We started with the 'easy' languages, French and Spanish, before moving on to Italian and Dutch, German, Polish, Japanese, Chinese, Arabic, and so on. We have used a variety of commercial and non-commercial engines, both rules-based (RBMT) and statistical (SMT).

In 2012 Lexcelera opened LexWorks to be the North American bridge to MT services such as consulting, creating engines and post-editing.

Over the last six years we have delivered major localisation projects using RBMT to shave off 50% of the time and 30% of the cost. (Our customers have achieved a positive Return on Investment – or ROI – from the first project, including engine training costs.) We have also delivered UI and courseware translations that were more highly rated for quality than their fully human versions. We have turned around 90,000 word translations overnight in raw SMT so that our customers could respond to calls for tender more quickly.

And what has all this taught us?

We have come to realise that machine translation is not a tool. Machine translation is a process. While much debate centres around the engine, whether rules-based or statistical, we have found that selecting the engine (or engines) is just one part of an industrial MT process – a process that, if correctly managed, is capable of lowering translation costs, increasing productivity and even improving quality and consistency.

To reach this goal, we believe that eight stages need to be in the MT process.

Pre-production
1. Consultation
2. Content
3. Customisation

Production
4. Piloting
5. Processing
6. Post-Editing

Post-production
7. Metrics
8. Maintenance

The last of these stages, maintenance, is the virtuous circle that links post-editing feedback to further customisations and real-time improvements in quality. In this process, machine translation technology is augmented by automated quality controls on both sides of the production process (automatic pre- and post-editing) and sees ongoing improvements by correcting issues flagged by the post-editors.

This article will look at the first three stages in the MT process: Consultation, Content and Customisation, and how the virtuous circle of post-editing feedback supports quality MT output.

1. Why does quality MT matter?

Until now it seems that the market has been considering MT quality as an unreachable goal. Current attitudes can best be summed up as: "It does not matter if machine translation is poor, as long as the post-editors clean it up."

So who pays the price of badly-managed MT processes that result in sub-optimum output?

The truth is that it is usually not the customer who pays for poor MT. It is the post-editors. By the time MT output gets to the end customer, any major flaws will have been fixed. I would venture to say that when we rely on post-editors to fix bad MT, the process does not matter at all. It does not matter which MT engine is used and whether or not it is trained well, because in the end there is always one secret weapon to make things right: the post-editors.

The post-editors, hired to repair machine translation, are the ones who pay the price for faulty processes. They carry the full brunt of a badly-trained MT engine, or of an engine that does not fit the content thrown into it. Post-editors work as hard as they need to in order to make sure that the customers get the quality they were expecting, regardless of what shape their material was in when spit out by the MT engine. And they do this at a steep discount over their normal rates.

The problem with this system is that it is not sustainable. For one thing, every post-editor unfairly paid to fix bad MT errors means that there just may well be one less post-editor who will say yes to the next project. I believe that poorly-managed MT is the main reason that the pool of those translators who are willing to be post-editors is not growing as it should.

Respecting the time of post-editors is the first, and arguably most compelling, reason that companies should strive for quality MT output.

2. Tool wars

Rather than concentrating on the full process for optimising MT, most of the debate today is stuck on what engine to use. Discussions – sometimes heated and not always evidence-based – compare today's two dominant approaches, and the competing software built up around them. Camps tend to be divided between supporters of rules-based machine translation and those of statistical machine translation, the first, as its name suggests, relying on grammatical rules that have been hard-coded into language-specific engines and the second using algorithms to first parse the text, then recreate it in another language based on mathematical predictions of the most likely translation.

The tool wars rage on – in conferences and online. RBMT and SMT face off against each other, their individual merits mostly argued by vendors of one solution or another. In the end, both sides may tip their hat to hybrid systems – the rare hybrid systems that actually *are* hybrid, as well as those that are merely a bit of window dressing by one side or the other.

The endless RBMT vs. SMT debates attract converts to one side or another, depending on which vendor is getting the most face time, or seems the most credible. Open source, naturally, gets a lot of traction due

to the attractive pricing (free, if you do not count the learning curve.) Evaluations abound, but they are often supplied by vendors of one solution or another: in tests, a trained SMT engine may be pitted against an untrained RBMT engine, or vice versa.

These contests are also inevitably skewed by the fact that content types and language pairs tend to play a determining role in engine performance. One engine is better at French, another at Japanese, and both sides declare victory.

Behind the face-offs are the metrics – with names like BLEU, NIST, GTM – which may or may not concur with each other, let alone with human evaluations. Critics complain that the playing field is not level and that certain metrics favour certain approaches. In a presentation entitled Language Research at DARPA, submitted to AMTA, the Association of Machine Translation in the Americas, Joe Olive, Program Manager, wrote that the leading metric, BLEU, is "not sensitive to comprehensibility or accuracy" and "favors SMT.")[1]

Since the very metrics used to measure quality may contradict a human judgment on whether a particular translation is good or not, automatic quality evaluation tools are just as inconclusive in declaring a winner in the MT wars as the face-offs have been.

Hype can be another disruptive factor in this discussion. Software vendors make claims that can only be tested with an expensive pilot. Language Service Providers (LSPs) may only be familiar with one particular tool, and so cannot offer customers the engine that performs best with their content, file formats or language pairs.

No wonder the market is confused!

3. Consultation

The choice of MT engine – Lucy, Moses, Systran, Language Weaver, Asia Online, PROMT, Bing, Google, Apertium, Reverso, and so on – is important. But it is not the only aspect to consider in the consultation phase. Considering just what engine to choose is akin to assuming that all there is to translation memory (TM) management is choosing between MemoQ, Déjà Vu and Trados, without any consideration of how a tool

[1] Unpublished presentation given at the 7th biennial conference of the Association of Machine Translation in the Americas, Cambridge, Massachusetts, USA, August 8-12, 2006.

would be integrated, what data it would include, who would manage it, and so on.

Machine translation is much more complex than translation memory. But the process bears similarities. For example, when choosing your TM tool, you may have made a decision about what content might be suitable. Next you may have consulted with your internal staff about what expertise and what resources you had in-house, what systems the TM tool would have to integrate with and what functionalities were most important. You may have also looked at what external resources were available to help you set up the TM systems, or even to manage the process. Finally, you would have looked customising the tool by populating it with your content.

All these considerations, and more, apply to the MT tool selection process. So rather than "What engine should I use?" the question should be "How will I determine what engine and what process will give the best results?"

The next step in the MT process is to decide how you want to manage MT. Do you want an internal engine that is customised to your content, or do you plan to outsource all MT activities, from customising and processing to post-editing and maintaining?

If you intend to use the engine internally, what kind of resources do you have? SMT will require more processing power, and strong engineering; RBMT managed internally will require more language resources. If you are bringing in an external vendor, would you want them to manage part (e.g. just training the engine) or all of the process (from training to post-editing)?

A linguistic audit may conclude the consultation phase and allow you to make basic decisions about how you intend to manage your MT process.

4. Content

Content is critical in the MT process as well. What type of information you want to translate is important in deciding what engine to use: not all content, not all file formats and not all language pairs are suited to the same engine. With Lexcelera's projects I have seen that RBMT engines such as Systran, Lucy, Reverso, Apertium and PROMT perform best with "narrow domain" content. That is, content with set terminology that needs to be respected, as in the case of software documentation and technical manuals. 'Broad domain' content such as patents and forums and other user-generated content are better suited to SMT engines such as Bing, Google, Language Weaver and Asia Online.

As for languages, almost any engine will do a pretty good job on French and Spanish. But for Japanese and German, to name just two, RBMT does

a better job. On the other hand, the minute you stray out of the dominant languages, SMT is the right choice. In fact, for 'exotic' languages, SMT is the *only* choice since RBMT engines tend to cover only the top 20 or so languages.

Engine choice is not only based on what type of content you want to translate, but also on what you have already translated. Specifically what kind of linguistic corpora have you built up? For example if you haven't got millions of segments of both good quality aligned translations (bi-text data) as well as monolingual content, there is usually no point in even thinking about SMT. This is because SMT tends to be a blank slate that you train with your own corpora. SMT generally learns your terminology at the same time as it learns your language pairs. RBMT, on the other hand, is already fully stocked on grammatical rules for the major languages and simply needs to be customised on the right terminology for your domain.

Another factor to consider when choosing an MT engine is what format your content is in. Content with tags that need to be protected usually is better managed by an RBMT engine, which can preserve the tags and not translate them.

The final consideration with content is how to prepare your text upstream so that it behaves better in MT: using software like Acrolinx to correct errors and inconsistencies, getting your writers to write with MT in mind by limiting sentences to 23 words and one single idea, and so on.

Once you have consulted with your internal teams, assessed your needs and external resources, analyzed and optimised your content, the next step is customising your MT engine.

5. Customisation

When we talk about customising or training an MT engine, the approach is different for a rules-based system and for a statistical system.

Google, Moses, Asia Online, Bing and Language Weaver are all examples of statistical engines. Systran, Lucy, Reverso, PROMT and Apertium are all examples of rules-based engines.

Not all these engines can be trained, of course. Users need a special arrangement to make more than limited improvements to Google and Bing. But all the others can and should be trained either by the customer or at least on the customer's content.

Training an SMT engine means training it for both the language pair - say, from English to Spanish - and for terminology. Both language and terminology are extrapolated from the content fed into the engine. That

is, each SMT engine is truly individual, based on algorithms that analyse what is found in the training data. This data will be made up of a minimum of one to two million segments of bilingual text (for an 'easy' pair such as English-Spanish) up to 20 million aligned segments (for Japanese, for example). The system also relies on target language monolingual data to complete the language model. Terminology is learned on this basis, using the probability model: the choice it has most often seen is the one that will be adopted.

In the same way as translation memory is language-independent because it sees only matching pairs, SMT can be trained for virtually any language pair, as long as enough bilingual content exists to train it on. This is the beauty of SMT. An engine can be created for any language under the sun, as long as there is a sufficiently large corpus of bi-text data to train it on.

The same cannot be said for RBMT. Rules-based engines are created for a specific language pair, with grammatical rules hard-wired – such as the noun-adjective word order in French and adjective-noun order in English. On the upside, RBMT comes ready to handle set language pairs right out of the box so even untrained there is a basis on which it can be deployed. The same is true of engines like Bing and Google, of course, which are already trained on an impressive number of language pairs.

The critical phase of RBMT training concerns terminology. While for SMT, terminology is learned at the same time as the language model is built, and relies on hefty data crunching, RBMT learns terminology from translation memories (the Systran hybrid uses TMs to build a language model) as well as from human linguists who encode glossary entries so that the system will understand them and always apply the right terminology.

This terminology knowledge can be built up per customer, per domain or per product line. In an ideal world, and for best MT results, the terminology would be at the most granular level possible. We train per domain (e.g. IT), per customer (e.g. for a particular company) *and* per product line.

If we favour rules-based engines for uses like documentation, On-Line Help (OLH), eLearning courseware and even User Interface (UI), it is because of the ease with which the engine can be customised to specific terminology, and the reliability of RBMT in returning the correct term.

Of course SMT can be customised to terminology. In the training phase, millions of segments of bi-text are fed into the engine. However, it is hard to control all the terminology choices that might be accounted for in all those segments. Since the user cannot control what the algorithms learn, they may sometimes learn the wrong thing.

For example, if multiple product lines, with differing terminology, are contained in the SMT training data, the engine will choose the terms that appear most often in the training data, whether it is the right term in this context or not. This can be a problem where a particular translation concerns Product Line A while most of the training data concerned Product Line B; in this case, the terminology of Product Line B will override the desired terminology.

Customisation of an RBMT or hybrid engine involves data mining and extraction to assemble lists of terms and their approved equivalents. This customisation is more than the simple glossary preparation needed to ensure that a human translator will have all the correct terms at his or her fingertips. RBMT customisation involves linguistically coding glossary entries as well as terms not to translate, such as a product name, a division, or a proper name.

To measure the impact of customisation in a rules-based hybrid, and at the same time determine whether another activity (controlled authoring) is more or less important than customisation, we carried out an informal study with John Kohl of SAS Institute. The content for the study was SAS Online Help documentation in HTML format and the engine was the Systran 7.0 hybrid.

In the first place, we compared the MT output we obtained 'out of the box' with that of a customised Hybrid engine. We could have used any number of measures such as BLEU or NIST, but coming from the perspective of a working translation company we decided to use a measure that was important to the cost structure of a translation: how much post-editing time was required to complete the task of bringing the two versions to human-quality.

The first measure we obtained was the baseline. That is, on that particular text, a translator could translate at the rate of 2400 words per day.

Next we obtained the post-editing speeds. We found that out of the box, the untrained engine generated output that could be post-edited at the rate of 4000 words per day. Next we compared that with the trained engine and discovered that customising Systran nearly doubled post-editing speed, to the rate of 7400 words per day.

The next task was to measure the impact of controlled authoring. To do this we pre-edited the text using Acrolinx rules. Sentences were shortened and simplified, as below:

Original:	**Acrochecked:**
Understanding the differences between owned and checked out alerts is critical to understanding SAS Anti-Money Laundering.	In order to understand SAS Anti-Money Laundering, you need to understand the differences between owned alerts and checked out alerts.

Table 1. Example of pre-edited text using Acrolinx rules

This study found that pre-editing source content improved post-editing time by nearly a third, to around 9400 words per day.

In subsequent studies as part of an EU R&D project with Moses, we were unable to duplicate the same high impact of controlled authoring on a statistical-based output.

6. The virtuous circle of post-editing feedback

There are those who consider post-editing a similar act to what translators do when they work with translation memory. That is, they believe that a post-editor's job is simply to correct the text. We do not believe that.

For us the post-editor is an integral part of the MT process. In our engine training for example, a crucial step after preparing the dictionaries that customise the engine is testing the output with a post-editor who not only corrects the mistakes, but makes those corrections directly in the engine, in real time.

Feeding back the corrections into the engine is the critical step in our MT process, where corrections from the post-editing phase are fed back into the system to improve the output. The ideal is to do this in as close to real time as possible in order to achieve maximum benefits from the post-editing process.

Post-editing should not be regarded as a stand-alone activity. While some corrections will be one-offs – fixing an error that is not likely to recur – at least 50% concern errors that should be fixed once and for all in the engine.

In our experience, the ideal process for a high volume project such as a software manual is to process a batch for the post-editors, get their corrections, then input another batch after their corrections have been input into the engine. For the first couple of years with a particular product line, the improvements tend to be the most dramatic; after that the improvement will continue with each iteration, but at a slower pace.

With SMT the virtuous circle of post-editing feedback and customisations is similar, although for most engines the corrections are fed into the system just once or twice a year, when enough data has been accumulated to justify a new training cycle.

7. Conclusion

Machine translation is the most exciting technology to come along since translation memory first made an appearance in the translation industry. If properly managed, MT can result in higher productivity and lowered costs, faster throughput and even improved quality and consistency. However, since these results are harder to achieve than similar results with translation memories, many in the translation and localisation industry have become stuck at just the first step in the MT process: choosing the tool.

But machine translation is about much more than what tool you use, whether rules-based or statistical. Machine translation is an industrial process. And, like any process it has inputs and outputs and a sequence of steps that need to be followed. A consultation phase helps lay the ground rules for how the process will be managed. In the content phase a closer look is given to the types of documents and languages that will be involved. Once the key decisions are made, the customisation phase involves customising – or training – the engine(s) on the specific types of content to translate. This is a critical phase to achieving MT quality, an important goal because post-editors run the risk of being burned out by amateurish MT.

Post-editors are essential to achieving human quality translations. But they can play a more active role than merely correcting what comes out of the MT engine. Post-editors can also have a major impact on engine quality through their corrections. This is the virtuous circle of post-editing feedback. In a rules-based process, corrections may be fed into the engine and the output improved at any point. In a statistical process, corrections are gathered in the form of translation memories that, once improved, can be fed back into the engine with the next data training cycle.

Properly training your MT engine and continuing to update it are two of the most significant activities for improving MT quality. We can say this with certainty based on the quality ratings our customers have given. A recent ratings report from Bentley Systems included the line: "Contrary to all expectations, using MT has improved the translation quality." Our score on that particular project for Online Help (in French) was 8.25/10 for MT, while the traditional TM translation was 8/10. For the courseware, the German reviewer for commented "It was nearly 9 […] it was the best translation of courseware I ever read."

Biography

Lori Thicke is founder and CEO of LexWorks, a subsidiary of Lexcelera (Eurotexte Group), established in Paris in 1986. Lori also founded Translators without Borders, the world's largest community of humanitarian translators. Lori, a Canadian, holds a Master of Fine Arts degree from the University of British Columbia and has been awarded prizes for her writing from the Canada Council and the CBC. Lori writes, blogs and gives presentations on MT technology and on the power of translation to unlock access to knowledge.

lori.thicke@lexworks.com