



## The Translator Must Always Be the Boss!

**National** Transportation Safety Board (NTSB) Chairman Christopher Hart said this about the investigation into the 2013 crash of an Asiana Airlines plane in San Francisco: “... we have learned that pilots must understand and command automation, and not become over-reliant on it. The pilot must always be the boss.” When I heard this on the radio this week, a big smile appeared on my face. What a perfect illustration of what I would like to share about translators and machine translation (MT)!

Recently, I was invited to represent translators as a speaker at the European Association of Machine Translation conference ([www.eamt.org](http://www.eamt.org)), and I am happy to report that it was a very rewarding event. One of the points in my presentation was a call to put an end to the assumption that post-editing is the only and most productive way to use MT.

You see, I think that we have looked at MT from only one angle—that of post-editing—and judged its effectiveness and desirability from that single perspective. It is no wonder that the majority of professional translators have not embraced post-editing of MT (PEMT). In the PEMT process, the proposed MT is the driving force; it is the agent (or, to use the NTSB chairman’s words, “the boss”), and it objectifies the translator or post-editor, who is reduced to a purely reactive role.

One reason I have been pushing the term “translation environment tool” (TEtT) is that it places the translator in the very center of a process supported by the tools he or she needs in order to work on and finish the translation successfully. There really is no place for post-

editing in that kind of environment. Why? Because post-editing merely reduces the translator to another tool.

So, what other ways could there be to use MT? I outlined a few of them as predictions in the May 2012 issue of this magazine, and now—two years later—they seem to be coming to fruition.

One that I have mentioned a number of times is using MT to correct fuzzy translation memory (TM) matches with MT proposals. This is a simple process that can prove to be very powerful. The idea is that if your TEtT can recognize which part of a translation unit is causing the match to be fuzzy (i.e., the “offending” part), and can identify which part in the target unit corresponds to that, then the tool will be able to go to an associated MT engine, get a proposal for the term or subsegment, and replace the “offending” target part.

Figure 1 on page 27 provides an example where the fuzzy match (*Die Augen der Katze sind braun*) was turned into a perfect match by automatically identifying the “offending” part (*braun*) and automatically replacing it with the correct term that it pulled from an MT engine. Since, in

this case, there were several MT engines connected to the project, there was actually more than one proposal. This becomes apparent when we see that *gelb* has been underlined.

Chances are you will still have to do some cleanup after that (or you might not be able to use the MT insertion at all), but you can see that this has the potential for a great deal of productivity improvement. In fact, if implemented well, it might even be possible to lower the fuzzy threshold (which most of us probably have at around 85% at the moment) to benefit more from TM content. Tools that presently support this include Déjà Vu X2/3 (this is the tool I used for the example in Figure 1), Cafetran, and Star Transit NxT, but the makers of Wordfast Classic/Pro, XTM, and Wordbee are also working on implementing this feature. (I think if someone were able and willing to implement the same in a Trados Studio app, it would quickly become one of the favorite downloads.)

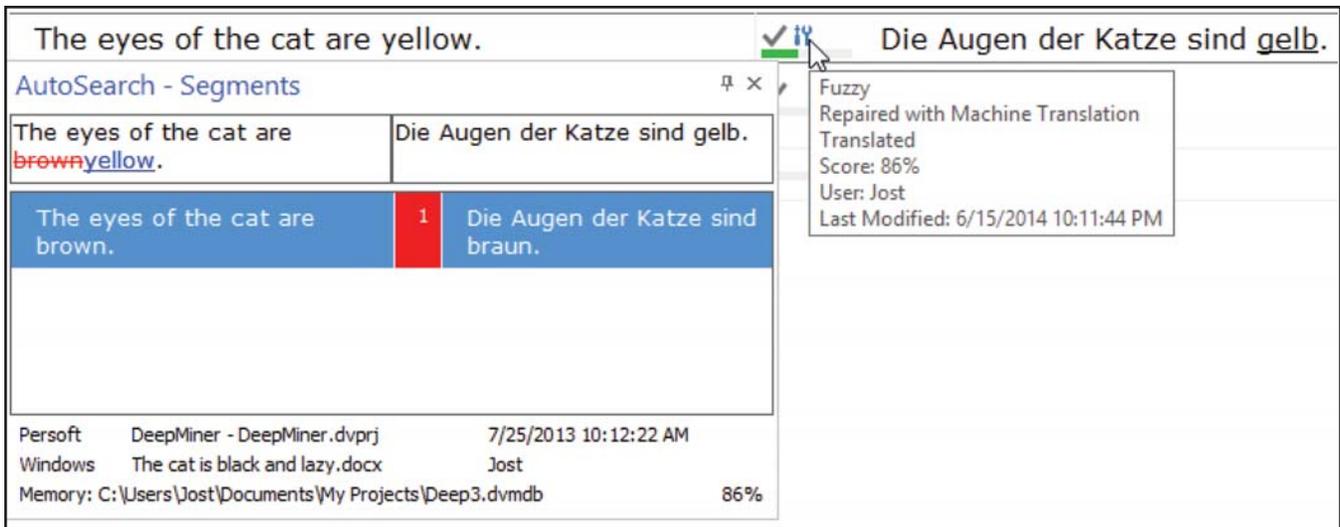
This is just one example of how MT can be used more productively and in a more translator-centric way.

Another translator-centric way of using MT that I have finally understood as a potentially highly produc-

## Information and Contacts

The GeekSpeak column has two goals: to inform the community about technological advances and at the same time encourage the use and appreciation of technology among translation professionals. Jost is the co-author of *Found in Translation: How Language Shapes Our Lives and Transforms the World*, a perfect source for replenishing your arsenal of information on how human translation and machine translation each play important parts in the broader world of translation. Contact: [jzetzsche@internationalwriters.com](mailto:jzetzsche@internationalwriters.com).

**Figure 1: Example of using MT to correct fuzzy translation memory matches with MT proposals**



tive way of using MT is the AutoComplete (or AutoWrite or AutoSuggest, depending on the tool’s nomenclature) feature that Wordfast Classic, Déjà Vu X2/3, and Trados Studio are already offering (in the case of the latter, through some apps from SDL’s OpenExchange app store).

This is how the process works. Rather than presenting MT proposals that need to be post-edited, it is the translator’s keystrokes that prompt suggestions from MT engines—very much along the lines of the translator as the driving force. If, as in the case of the example presented in Figure 2, there is more than one MT engine associated with the project, all of

them are polled interactively for matches with the keystrokes that have already been entered. The suggestions are displayed as an AutoComplete tool tip that you can select by pressing the Enter key or with a mouse click.

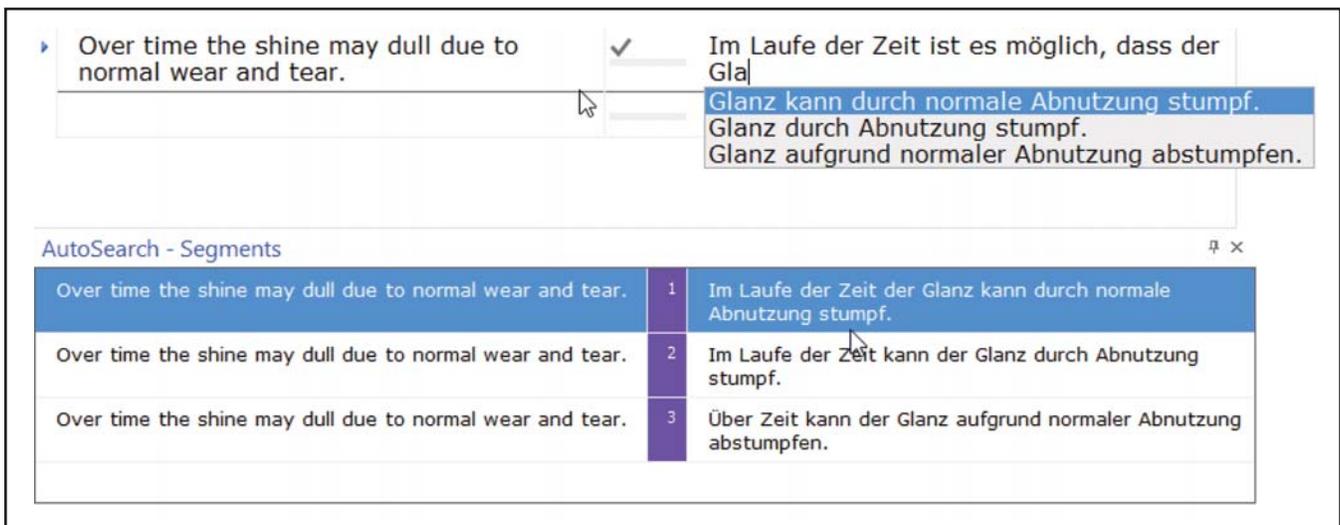
This is particularly interesting because it shows more than just the complete matches for the whole segment as you translate. It also shows the subsegments, which continue to adjust themselves to what you type. Now, for those among you who read German, you will recognize that none of the suggestions presented in Figure 2 are perfect as a completion for what had already been entered, but the

third suggestion comes very close to a possibility, with only the need to change the ending of the verb.

I recently finished a large project where I tested this way of working with MT. I really liked that it saved a lot of time while also avoiding the unfortunate influence of preoccupying the translator’s mind with suggestions (a process with which most of you are familiar from TM and MT). In fact, I found myself more often than not just choosing a few words and phrases here and there from MT suggestions rather than lengthy segments. Still, I know that I was able to work significantly faster.

The example in Figure 2 was done

**Figure 2: Polled MT suggestions displayed as an AutoComplete tool tip**



using Google Translate, Microsoft Bing Translator, and Systran, but I realize that there is so much more that can be done with such a system.

Here is a case in point. While the European Patent Office (EPO) has chosen to collaborate with Google to build up a version of Google Translate for patent translation ([www.epo.org/searching/free/patent-translate.html](http://www.epo.org/searching/free/patent-translate.html)), the World Intellectual Property Organization (WIPO) strongly rejected that concept and built its own MT engine based on the open-source Moses platform. So far there is only a handful of languages available (English into and out of French, Japanese, Chinese, and German), but it has some features that are really interesting. Especially when you go to

its “interactive” edition, you can see a couple of the more advanced features (<https://www3.wipo.int/patent/scope/translate/tapta.jsf>).

In the lower half of the window (which appears once you send a text for translation), you can see that when you select any part of the source segment, a translation for that subsegment is entered into the target pane. Of particular interest, the MT engine not only gives you that one match, but also a list of other possible matches in the dropdown box right under “Proposals.” That is how MT works. For any segment or subsegment, a very large number of proposals is generated, but typically only the one with the highest (internal) confidence score is actually used and exposed to the user. The WIPO developers have managed to show us other matches as well.

It would be interesting for translators who want to use a translator-centric MT process to have all of the possibilities—or at least some of them—displayed as AutoSuggestions. The latest incarnation of Google Translate does something similar. While providing the translation for the whole segment, you can click on individual subsegments to see alternate translations. To some degree, Xingzeng Liu, the developer of the free Trados Studio app Google Translate AutoSuggest uses this mechanism (<http://bit.ly/google-auto-suggest>). After you install the app, it looks for your first keystrokes and suggests the translation of the complete segment from Google Translate if it matches the characters you have entered. If your keystrokes do not match the Google Translate proposal or you choose not to use it, it will suggest subsegments according to the subsegmentation that Google Translate has already performed. Unfortunately, it does not also suggest the alternate translations, which would be very nice.

If you do use Google Translate as a translation aid and are a Trados

user, you should definitely think about using Xingzeng’s tool. He has found a way to retrieve the Google Translate matches without forcing you to pay for the application programming interface access (enjoy this while it lasts).

While we are talking about Trados AutoSuggest apps for MT access: CodingBreeze’s MT AutoSuggest app is interesting as well (<http://bit.ly/MTautosuggest>). This works with any MT engine with which you might have associated your project and gives you AutoSuggest proposals for subsegments. It bases the delimitation of those subsegments on things like segment-internal punctuation (including commas, semicolons, etc.) and a certain number of words. It does not yet propose suggestions from more than one MT engine at a time, but I talked to the developers and they promised me that they would work on that next.

So, to come back to the purpose of this column, these two translator-centric ways of using MT have already been implemented in some commercial tools, and others will soon follow. (Laboratory tools such as CASMA-CAT or MateCat are also experimenting with more advanced MT features, but they are not really in the production phase.) There is plenty that can be done better, and there are plenty of other more creative uses of MT, but what is important in all of this is that we need to stop talking about post-editing as if this were the only way to approach MT.

I am willing to bet that with a tool that has a well-implemented AutoSuggest feature for MT or several well-trained MT(s), professional translators can produce high-quality output in a more productive manner than by post-editing raw MT output from one engine. And they can do that while respecting and utilizing their human translation skills and environment. ■

## Education On-Demand ATA eConference Available Now!

Couldn't attend ATA's Annual Conference in San Antonio? Check out the online eConference—a total of 162 presentations and pre-conference seminars in a webinar-style format. It's a convenient and flexible way to learn.

- More than 182 hours of educational content
- Unlimited online access
- Webinar-style format
- Handouts and support material
- MP3 files for download

### Earn ATA Continuing Education points!

- One continuing education point for each hour viewed (maximum 10 points)

For more information, go to [www.atanet.org/conf/2013/econ.htm](http://www.atanet.org/conf/2013/econ.htm).