BY JOST ZETZSCHE

Talking with the Developer of GT4T

Dallas Cao is the developer of GT4T[1], a little, unobtrusive application that allows you to connect from any Windows or Mac application to a large range of machine translation (MT) engines. As such, it represents a different way of accessing MT suggestions from many of the translation environment tools within their interfaces, and it also allows you to access MT suggestions from within any non-translation-specific environment. Also, you can use GT4T to automatically override terminology used by the MT engines. I talked to Dallas about the history of the tool, its features, and his future plans.

Jost: More and more translators use MT as one of their resources for translation. Professional translators who are using a translation environment or computer-assisted translation (CAT) tool usually use an API-based[2] connector to a MT engine that brings the MT suggestions right into their environment alongside translation memory matches, term base suggestions, and other resources. Your tool, GT4T, deals with MT differently. But before we get into what it actually does and how it can be used, tell us a little bit about GT4T's history and why you chose to create it in the first place.

Dallas: The original idea of GT4T is simple. You select a portion of source text and press a keyboard shortcut, and the selection is then replaced by Google's MT translation.

I started working on GT4T for my personal use as early as 2009, when the neural MT engine was nonexistent and MT was little more than a laughingstock. But I found while MT was almost always bad at understanding the structure of a sentence, it could be used to translate phrases. I wanted to have a tool that would allow humans to decide and choose which part of a sentence to be "translated" by MT on the fly without disrupting their workflow.

I was a translator who had never thought of becoming a programmer. If there had been such a tool then, I would have been a happy user of it and there would never have been a GT4T.

I wanted to have a tool that would allow humans to decide and choose which part of a sentence to be "translated" by MT on the fly without disrupting their workflow.

The first version of GT4T was written as a Microsoft Word macro and only worked in Microsoft Word. I was excited to find that it was even more useful than I had initially thought, and very soon the idea of selling it came to my mind. There are always phrases that human translators know MT will certainly do well with, like a list of country names. Using GT4T would simply save some keystrokes. With time, that alone would be a huge productivity gain.

To make a long story short, the spirit of entrepreneurship is to continually push a simple idea forward and see how far it can go. With some twists and turns, GT4T has grown in features and translation quality. As MT gets better, GT4T automatically gets better too! I have also grown into a confident programmer.

J Let's talk about the tool itself. It runs on Windows and Mac and gives access to Google Translate (either the neural or the statistical MT engine), Microsoft Translator, DeepL Pro, Yandex, and a variety of Chinese-based providers, including Baidu, Youdao, Tencent, Sogou, CloudTranslation, and NiuTrans. The user can select which engines—and which language combination—they want to use, and upon highlighting text in any application and pressing a keyboard combination, the result(s) is displayed in a pop-up window. If any of the suggestions are helpful, they can replace the original text in the originating application. Am I correct so far? Do you want to talk about some other features that differentiate GT4T?

D GT4T also offers special shortcuts that allow you to automatically translate segments in a CAT tool like SDL Trados Studio or memoQ. You can hit a shortcut to translate the current segment or several segments. The shortcuts work in a long list of CAT tools, including web-based ones like Smartling and Crowdin.

Other than MT engines, GT4T also helps access various online dictionaries in the same fashion—without having to leave your working environment or open a browser window. You can use a shortcut to submit your selection simultaneously to several online dictionaries, glossaries, or terminology sites like the Interactive Terminology for Europe (the EU's terminology database), Microsoft Glossary, or the terminology collection at Proz.com. GT4T goes one step further than similar tools like IntelliWebSearch. Instead of automatically opening the webpages, GT4T collects the dictionary results and displays them in a pop-up window. The user can pick a translation and hit Enter to insert it into the document in which they are working.

J One interesting feature of GT4T you didn't mention is the custom-made glossary that automatically replaces terms in the MT suggestions. I assume that it's particularly valuable for languages with no morphology, like Chinese. What about languages with rich morphology (which I assume will result in a lot of missed replacements)? Can the user apply some kind of wildcard to find morphological variants? And is there a way to accommodate things like gender and possible automatic replacement of articles or pronouns?

D The replacement feature doesn't support wildcards, just exact matches, and there's no way to accommodate things like gender, nor are there any future plans for this. A feature like that would require a team of linguists who know many languages. So far, GT4T is still a one-man endeavor, and I have no plan to have a team or incorporate it.

I have limited knowledge of linguistics and don't know how well the replacement feature would work with language pairs other than English>Chinese. However, I know of at least one Dutch>English translator who is very excited about this feature, so I assume it's also useful for languages with rich morphology when properly used.

Other than replacing MT translation results, users can also use glossaries to keep their translations consistent. The glossary file is an Excel spreadsheet and users can easily import old translations or add items. To find out how a term is translated in previous translations, you'll just have to select it and press a shortcut. And you're not limited to one CAT tool or app. You can search the glossary anywhere in any app.

J I personally think that MT as a translation resource is often more useful on a subsegment level. The way that's often done in translation environments is to have the tool automatically select subsegments from a longer MT suggestion on the basis of keystrokes. Is that something that's possible with your tool? And if not, are you thinking of introducing something like that?

D Yes, that's actually the original idea of GT4T! You select a chunk of text, whether it be a phrase or even a word, where you think MT will do a good job. I always want GT4T to be used only as a productivity tool, a reference. Translators use it to save a few keystrokes, or get translation suggestions on a subsegment, or simply when their brain stops at a word and needs to be nudged.

As the sole developer of this app, my strength is flexibility, but I also know my limits, which is why I've been hugely dependent on users' suggestions and reports for new features.

J But that's not really what I mean. What I mean is rather than manually highlighting a fragment and looking for a translation, it's much more efficient to have various MT suggestions in the background that display fragments only when there are matches between the first few keystrokes of the translator within their translation environment and something within the MT suggestions. I understand that this is possibly not as relevant in a target language like Chinese, but it is for many other languages. Is that something that could be implemented?

D That's an interesting idea, and would be revolutionary if this could be done. GT4T is a standalone app that offers system-wide keyboard shortcuts. It cannot "see" the text of a document until a user makes a selection and hits a shortcut. Perhaps it's easier for CAT tool developers to implement this feature within their tools. It could also be an add-on. As a standalone app, GT4T would probably need to pre-translate documents first in the background. I'll think about it.

J In your tool, it's possible to enter personal API keys for the various MT engines. I assume this means that I would then receive suggestions from that respective engine via my own API and pay for it. How is that different from receiving suggestions without me entering an API? Can users access their own customized engines at Google or Microsoft like this? I assume that you're also accessing the engines via your API key and that you need to pay for that. How does that work for you financially? Also, some users who use your tool to access DeepL Pro cannot access it otherwise because they're living outside the European Union. Suggestions from DeepL Pro are more expensive than those from other tools. How do you account for that?

D Yes, users can choose either to use their own API and pay a small subscription fee for GT4T, or buy a plan that already includes MT data. There's no difference between using your own API or the built-in API. You get the same results from the MT engine. I haven't started on the customized engines yet, but I'll certainly study them when I have time.

GT4T offers very flexible plans. Users can buy either time-based packages that have no usage-limit, or character-based packages with no time-limit. I pay MT engines on the basis of the number of characters used. Time-based licenses work on only one computer at a time, but you can install a character-based license on up to 30 computers. I calculate profits for each purchase using server-side scripts. Occasionally I do actually end up paying more for a user than they pay me, but on the whole I make money from time-based licenses. But the danger is real: theoretically, a very hard-working translator on a time-based plan could bankrupt me, and I don't have a plan for that.

GT4T is also valuable as a free tool. The dictionary, glossary feature, and two MT engines (Yandex and Tencent) are actually free.

J Let's talk about the accessible engines I listed earlier. What I'm missing are engines such as Amazon Translate, SDL BeGlobal, PROMT, and Naver Papago. How do you decide which engines to include, and can users ask you to add engines?

D The first thing I consider is translation quality. It seems DeepL and Google excel in quality, and there's no urgency to add more engines. I seriously consider suggestions from users. Some engines don't offer API access. I recently added Systran at the request of a user and later removed it because I couldn't reach an agreement with Systran. I recommended NiuTrans to her and she was happy with it. By the way, NiuTrans is a dark horse that deserves more attention. It performs pretty well, even for European language pairs.

J I'm not sure I agree completely. For instance, in the case of Naver Papago (whose engineer we interviewed in the November/December issue[3]), the results are often judged better by Korean users. Amazon also might produce better results in some language combinations. Maybe it would be possible to just add the framework so users can add their own API keys for some engines that are not supported?

D Thanks for updating me with this information and recommending Naver Papago and other engines. If they offer API access and I can reach an agreement with them, I'll certainly add them to GT4T. Users certainly will then be able to use their own API as well.

I always want GT4T to be just a productivity tool, a reference.

J Here's a question about security and privacy. You're located in China, which might be a concern to some users or their clients. Are requests to the various MT engines actually visible to you and/or do you store that data? Or does your tool just facilitate the connection between the user and the respective MT engine so you don't actually ever get to the data?

D Neither the requests nor the replies from MT servers are visible to me or are being collected. When a user selects some text and presses a GT4T shortcut, the selection is sent directly to the respective MT servers and then the user receives replies from them. GT4T collects the following for licensing purposes: a unique hardware code and the number of characters you submit to MT servers through GT4T. The last time you used GT4T and your IP address are also collected automatically by the GT4T licensing server. They're actually collected by all websites you visit, and it takes hard work not to collect them.

J Any future plans with GT4T that might be interesting for us?

D I'm currently working on a new version that does document translation. It will support many file types, including popular CAT file types as well as Microsoft Office types. Users will then be able to translate their documents without having to upload them to a server, and they can even browse a folder and ask GT4T to translate all the files in the folder and subfolders in the background. Your idea of providing suggestions while a user is typing is also helpful. I'll reevaluate it and other suggestions that you brought up.

As the sole developer of this app, my strength is flexibility, but I also know my limits. Frankly, I've been hugely dependent on users' suggestions and reports for new features, and sometimes even for debugging. ○

Remember, if you have any ideas and/or suggestions regarding helpful resources or tools you would like to see featured, please e-mail Jost Zetzsche at jzetzsche@internationalwriters.com.

NOTES

[1] You can check out GT4T at https://gt4t.net/en/.

[2] API stands for application programming interface, and is the technology that allows different programs to talk with each other (such as Trados or GT4T with Google Translate).

[3] Zetzsche, Jost. "Thoughts on Naver Papago with MT Engineer Lucy Park," *The ATA Chronicle* (November/December 2019), 30, http://bit.ly/Naver-Papago.

Dallas Cao is an English>Chinese translator and self-taught programmer. He developed GT4T, a Windows/Mac app that allows users to use online machine translation and dictionaries in any programs without having to open the browser. Contact: dallascao@gmail.com.

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.