MTradumàtica: Statistical Machine Translation Customisation for Translators

Adrià Martín-Mor

Abstract

This article presents the results of the research project ProjecTA, which attempts to bring machine translation within closer reach of translators. This works from the basic assumption that the translator's profile is valid for managing MT-related tasks. The first phase of this project consisted of researching to what degree translation agencies in Catalonia and Spain use MT, via a survey sent to translation companies. The results showed that MT is used very little by micro and small enterprises, which led to the second phase of this project, namely to develop a free MT platform, MTradumàtica, specifically designed for translators.

Keywords: Machine Translation, Statistical Machine Translation, Machine Translation Customisation, Moses.

Introduction

Machine translation (MT) is radically and rapidly changing the way humans deal with languages. The world of translation – professional translators, universities and translation agencies – is struggling to adapt to these changes; some simply disregard MT, while many others see it as an opportunity. ProjecTA (www.projecta.tradumatica.net)[1] is a Tradumàtica group research project (www.tradumatica.net) at the Departament de Traducció i d'Interpretació i d'Estudis de l'Àsia Oriental at the Universitat Autònoma de Barcelona. It works from the basic assumption that translators have the appropriate profile to manage MT-related tasks, and that empowering translators in MT tasks is beneficial for translation companies. The project was split into two phases: first to explore how MT is used by the translation sector in Catalonia and Spain through a survey sent to 187 translation agencies; second, based on the survey responses, to develop a free MT platform, MTradumàtica.

This article is divided into seven sections. Section 1 (*Types of MT*) provides an introduction to MT and the main types of engines that exist nowadays. Section 2 (*ProjecTA and the use of MT in translation companies*) presents the methodology for this project, focusing on the answers to the survey. Section 3 (*Rationale for an MT platform*) justifies the need for an MT platform which fulfills the needs of the local translation industry, while section 4 (*Statistical Machine Translation Customisation*) focuses on the available software that allows customising statistical MT. Section 5 (*Description of MTradumàtica: Interface and processes*) describes the MTradumàtica interface and the processes that it allows, with references to the concepts presented in the previous section. Section 6 (*Future developments*) presents some of the functionalities that are envisaged for MTradumàtica, ending with Section 7 (*Concluding remarks*).

1. Types of MT

Depending on the technology used, MT systems can be broadly classified using the traditional distinction between rule-based (RBMT) and corpus-based MT (Ping, 2009: 162).

RBMT uses dictionaries as well as rules —lexical, transfer and, occasionally, semantic rules— which are written specifically for each language pair. This means that it is a rather expensive technology in terms of human resources required to create the necessary linguistic data. Although dictionaries and rules written for a given language in a specific language pair might be re-used for other language pairs, most of the linguistic information has to be re-written for each engine. Nevertheless, RBMT provides a similar output in terms of quality as other technologies, especially between languages of the same family (Forcada, 2009: 13; Forcada et al., 2011).

The corpus-based alternative comprises three approaches: example-based MT (EBMT), statistical MT (SMT) and neural MT (NMT). Some of the most known MT systems are statistical (like Google Translate), while NMT opens up new research perspectives to bring about higher quality outputs.[2] SMT relies on corpora to statistically select the most probable translation for a given source sentence, and needs a bilingual parallel corpus and a monolingual corpus. These corpora are processed by the system in order to create what is known as the translation model (TM) and the language model (LM), respectively. The TM contains information on how sentences should be translated according to statistical probabilities, while the LM helps the system to select the most probable sentences based on real texts, and therefore ensures the fluency of the output. In contrast to RBMT, an SMT engine can be created in a relatively short time span: most of the work needed to create this engine is usually devoted to corpus cleaning, but once linguistic data is ready, the training process can be carried out in a matter of days. Furthermore, many SMT engines use pivot languages to increase the number of available combinations offered (e.g., translating from A to C via B). Because SMT relies entirely on corpora, this is not a viable solution in the case of languages for which there is no consistent collection of texts. Therefore, the vast majority of minoritised languages that are not fully standardised are *de facto* excluded from using this technology, since they often do not have large amounts of linguistic data available. As regards hybrid engines, these combine the aforementioned technologies and are typically either statistically-based with specific rules, or rule-based with statistical components.

At another level, MT systems can be generic or specific. According to Ping (2009: 162), customised systems "are targeted at groups of users who work in specific areas or fields (domains) … [and are] much more effective than generic MT." In other words, systems which have been tailored to the specific needs of a project by taking into account text domains will most likely produce better results. This article focuses on the use of SMT, particularly SMT customisation (SMTC), i.e. tailoring SMT engines to the user's needs.

2. ProjecTA and the use of MT in translation companies

The mission behind ProjecTA was to monitor introducing MT and post editing (PE) into the translation process and to propose ways to make translators and translation companies more competitive. Its ultimate goal was to prove that translators have the appropriate profile to

undertake tasks related to implementing and managing MT engines in translation companies. This assumption attempts to address what seems to be a well-established tendency in the translation industry today of seeing the translator profile as dispensable when it comes to technology-related tasks, so translators are often seen as mere users of MT. This in turn can reinforce negative views already held by some translators about MT, which often see technology in terms of lowering their rates (Torres-Hostench, Presas & Cid-Leal, 2016: 41). ProjecTA attempts to empower translators, which means taking on tasks such as preparing linguistic data, training the system, pre-editing, post-editing and modifying the system.

The first phase of the project consisted of analysing the use of MT and PE in the professional translation industry in Catalonia and Spain. To this end, a survey was created with 17 questions covering the following four areas:

(1) Details about the company

(2) Services and languages offered

(3) Clients

(4) Use of MT and PE

The survey was e-mailed to 187 companies between January and February 2015, from which we obtained 55 responses. Most were micro companies (<10 workers) or small companies (10-49 workers), of which the majority of them were based in Barcelona (23) and Madrid (10). According to the results, (Torres-Hostench, Presas and Cid-Leal, 2016: 16), almost half of the companies that answered the survey (26 – 47.3%) include MT in their workflow. However, a closer look at the results reveals that half of these use MT for up to a maximum of only 10% of their projects (2016: 21). This is consistent with answers received to the question on the percentage of projects devoted to PE, since half of the companies stated that only 10% to 20% of all their projects are PE (2016: 24). As for the typology of systems, it should be noted that customised MT engines are used very little among those companies that use MT (26): while one company did not respond to this question, 5 use SMT, 3 use hybrid MT and 2 use, RBMT, which makes a total of 11 companies using customised MT engines.

All these data led to the conclusion that MT use among Catalan and Spanish translation agencies is low. Some decisive factors which explain this were identified, both in the answers to the survey and in a focus group of translation companies contacted directly in June 2015 to obtain qualitative data, which were related mainly to either financial, training or confidentiality issues. Starting with financial concerns, companies are aware that implementing MT in their workflows is costly, not so much because of the software, but in terms of human resources: trained professionals must work on preparing linguistic data and carry out productivity tests, etc. This leads to the second concern, training, both of the person responsible for implementing MT as well as the trainee translators, who need to be trained how to post-edit and use the MT system. Furthermore this training also includes preparing linguistic material, such as PE guidelines. Finally, confidentiality is one of the biggest issues for companies: the confidentiality of texts has to be monitored throughout the translation process, preferably performed on the company's premises, and not re-used by third parties. Failure to do so may be in breach of

confidentiality clauses included in translation contracts.

The conclusions of this study provided the research group with the basis for the subsequent project phase, and in this second phase, ProjecTA set out to develop software that could remove some of the barriers to implementing MT systems in the Catalan and Spanish translation industry.

3. Rationale for an MT platform

There are already a few products for SMTC that allow the user to set up an engine based on their own linguistic data, such as proprietary systems like KantanMT, Microsoft Translator Hub or LetsMT, which also include some extra features such as automatic quality estimation or search and replace functionalities within the corpora. Moses, free software released under the GNU Lesser General Public License (LGPL),[3] is one of the most popular systems for SMTC. According to LT-Innovate (2013: 71), Moses is "widely used within the industry to build customized MT engines" and "[b]ecause it is open source, people wishing to develop a custom engine can focus on obtaining the training corpora rather than writing their own statistical machine translation engine (a difficult task that is beyond the abilities of most developers)." Although it is mostly used in GNU/Linux, it is multi-platform in that it can be installed in other operating systems (OS) by running compatibility layers (e.g. Cygwin for Windows) and package management systems (such as MacPorts for MacOS). Some researchers have developed software ―such as Moses for Mere Mortals― based on Moses with the aim of making it more accessible and user-friendly for non-experts (Machado & Hilario, 2014). However, as LT-Innovate (2013: 72) puts it, Moses is "difficult to administer". In other words, it requires some skills with command-line terminals and UNIX-like systems, and it does not have a graphic user interface (GUI).[4]

As mentioned in section 2 (*ProjecTA and the use of MT in translation companies*), the results of the survey carried out by ProjecTA led to identifying some obstacles to using MT in the Catalan and Spanish translation industry, mainly related to cost, training and confidentiality. Taking into account these concerns, Moses was deemed the best option to work with within the framework of ProjecTA. Firstly, because the Moses' LGPL licence allows users to modify the source code and redistribute the resulting software. Secondly, because it allows for decentralised installations ―meaning it can be installed on private servers―, and therefore confidentiality is guaranteed (insofar as the server is secure). Thirdly, because it is free software and users do not have to pay for it. From a research policy perspective, contributing to free software also means that the product of research projects financed with public funding is repaid to the whole society.

Nonetheless, as mentioned above, Moses has a steep learning curve, especially for translators, since it requires some knowledge of how to use command-line terminals and UNIX-like operating systems. It was deemed appropriate, therefore, to design a Moses-based platform specifically for translators. This platform was to be web-based and have a GUI, with a special emphasis on being user-friendly for non-experts. A GUI is a must for most translators, since they might not be familiar with the use of command-line terminals. Furthermore, having a web-based interface makes any software platform-agnostic, i.e., operating system independent. These considerations do not necessarily imply that the skills mentioned (such as knowledge on

command-line terminals and UNIX-like OSs) are not deemed needed or desirable for translators, but rather that they tend to deter translators from delving into MT.

Some research has been carried out in the field of post-editing to identify what is required of software in order to meet the needs of post-editors. As a result, there are already some programs specifically designed to post-edit, such as PET (Post-Editing Tool, see Aziz, Castilho & Specia, 2012).

4. Statistical Machine Translation Customisation

This section describes SMTC and its associated processes taking Moses as a reference.[5] As explained in section 1 (*Types of MT*), two corpora is all that is needed to create an SMT system: a bilingual parallel corpus (to create the TM) and a monolingual corpus (to create the LM). As a matter of fact, it is only partially true that two corpora are needed, since it is technically viable for SMT systems to build LMs from a single parallel corpus by training the LM on the target language of that corpus: "The language model should be trained on a corpus that is suitable to the domain. If the translation model is trained on a parallel corpus, then the language model should be trained on the output side of that corpus, although using additional training data is often beneficial." (Koehn, 2016: 256). Instead, TMs and LMs can be created from as many corpora as is needed.

Under Moses (as in other systems), corpora undergo some processes before they can be trained: tokenisation, truecasing and cleaning (Koehn, 2016: 36). *Tokenising* means separating words and punctuation with spaces. Tokenising texts allows you to isolate punctuation in order to increase match-finding probabilities with the future source texts that will be machine-translated. Truecasing consists of a two-phase process by which the most probable casing is assigned to the initial words in each sentence so that data sparsity is reduced. Cleaning is the process of removing long, empty and misaligned sentences from the corpora in order to minimise possible training problems.

Once these processes have been carried out, linguistic data is processed by the system, called *training*, in which translation correspondences are inferred between the two languages by analysing co-occurrences of words and segments (Koehn, 2016: 11). The result of the training process is the TM, which consists of a *phrase table* and an LM, and may contain a *reordering table*. While the phrase-table contains the statistical information of source-target equivalences, the reordering table describes the changes in the word order between source and target languages. Since these tables might be slow to load, they can be binarised, i.e., compiled into a format that can be loaded faster.

During the training process standard weights are assigned to the statistical models. These standard —and therefore non corpora-specific— values can be modified to improve the performance of the system, for instance, giving priority to shorter sentences. Therefore, after training, a process of *tuning* can be carried out, meaning that "the different statistical models are weighted against each other to produce the best possible translations" (Koehn, 2016: 12). In other words, the optimal weights for each parameter (which can also be modified manually) are sought by measuring the quality of the translation with a "small amount of parallel data, separate from the training data", called the *tuning set* or *development set* (Koehn, 2016: 39).

Tuning an engine means translating thousands of sentences with a TM, comparing them with the human reference translations in the tuning set and adjusting the weights assigned to each value in order to improve the quality of the engine, which means that tuning is a slow process. Quality is measured by using automatic evaluation metrics, such as BLEU (see Papineni, Roukos, Ward, & Zhu, 2002).

The objective behind developing the platform within the framework of ProjecTA was to allow users to train a complete MT engine. Therefore, familiarising users with the basic concepts and processes, such as those described above, could prove useful for those who want to further manipulate the system. For these reasons, as explained in the following sections, some references to the aforementioned processes appear in the GUI, especially in the advanced phases of customisation.

5. Description of MTradumàtica: Interface and processes

Taking the previous considerations into account, a Moses-based web platform with GUI, was developed: MTradumàtica. It is currently available for testing purposes and can be installed stand alone on a PC as well as on a server so that it can be accessed from any computer with an internet connection via a web browser. A compressed package is provided (less than 3 MB) with installation instructions, which will allow any user to install their own version of MTradumàtica.

From a pedagogical point of view, great importance was attributed to the training potential of the platform: the process of building up an engine was to be graphically represented, clearly showing each of the steps of the process. This decision to do this was based on the firm belief the user's control over the whole process would increase by making all the necessary phrases visible.

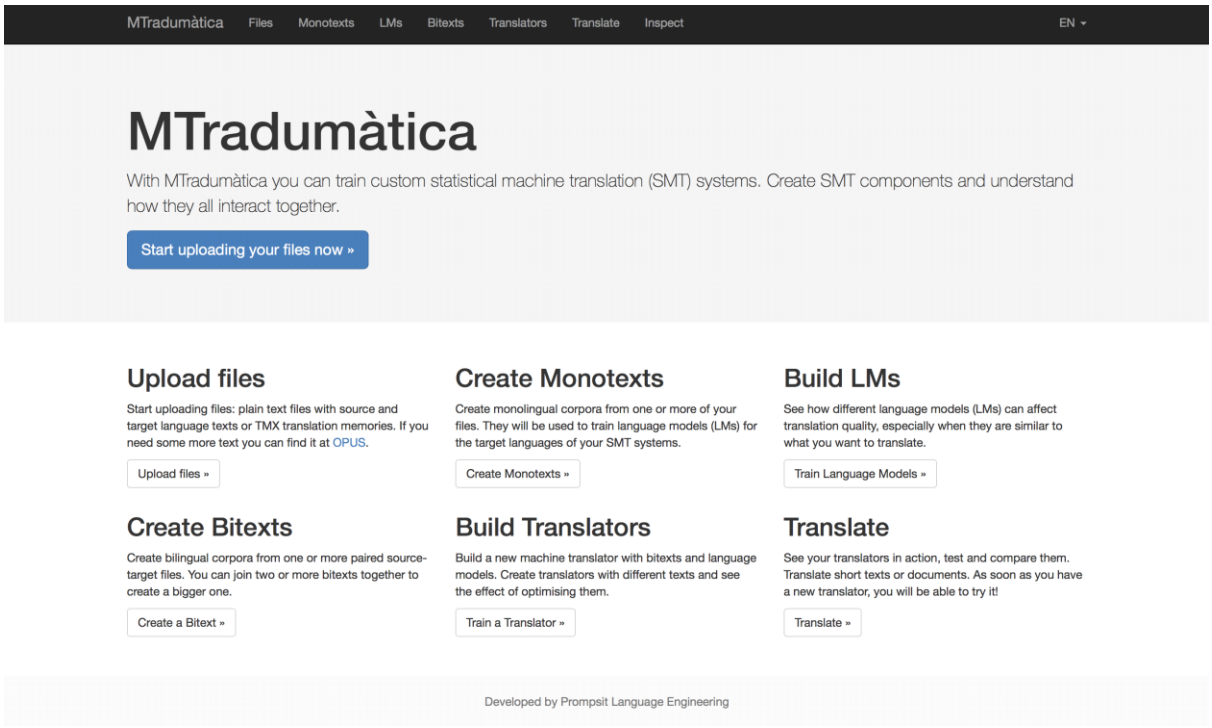

Figure 1 *Home page of MTradumàtica*

Since the MTradumàtica interface is designed so that users can build an SMT engine based on their own corpora, with an emphasis on making this easy to follow and understand, the current version of MTradumàtica breaks up the entire process into six sequential steps (given the same names in the interface for the sake of clarity):

(1) Upload files

(2) Create monotexts

(3) Build LMs

(4) Create bitexts

(5) Build translators

(6) Translate

The top bar (see figure 1) shows all the steps at any time during the whole process, and indicates at which stage their systems are. The simplified schema below explains how an LM (step 3) and a TM (step 4) are generated, which are finally combined to build a translator (step 5). For the system to be able to generate these models, users are first prompted to load their corpora by using a file manager (step 1). The following figure graphically represents this schema, alongside the underlying processes described earlier in section 4.

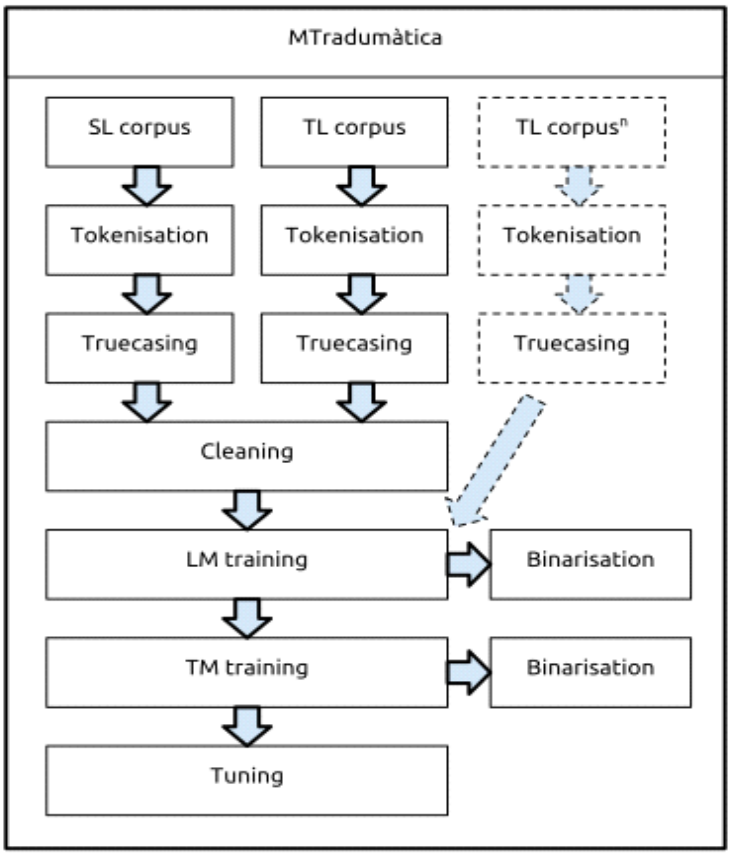

Figure 2 *Schema of MTradumàtica*

This schema describes how MT engines are created with MTradumàtica, based on at least one bilingual corpus made up of source language (SL) and target language (TL) texts and zero or more monolingual corpora (in TL) – see section 4, *Statistical Machine Translation Customisation*. Each corpus is tokenised and truecased, and cleaned in the case of a bilingual corpus. From the training process, an LM and a TM are obtained, which can be binarised so that they load faster. Then the last step is to tune the engine.

The previous schema corresponds to the aforementioned steps of the interface in MTradumàtica. The process starts with the File manager, which allows users to upload texts to the platform. Currently, MTradumàtica supports only text files with one sentence per line (Moses format). Although this may seem a rather specific format, it is quite widespread on the internet. The OPUS project[6] —which is linked from the MTradumàtica home page (see figure 3), hosts corpora for large number of languages coming from free online data, such as Wikipedia and Ubuntu. These corpora can be downloaded in several formats, one of them being the Moses format.

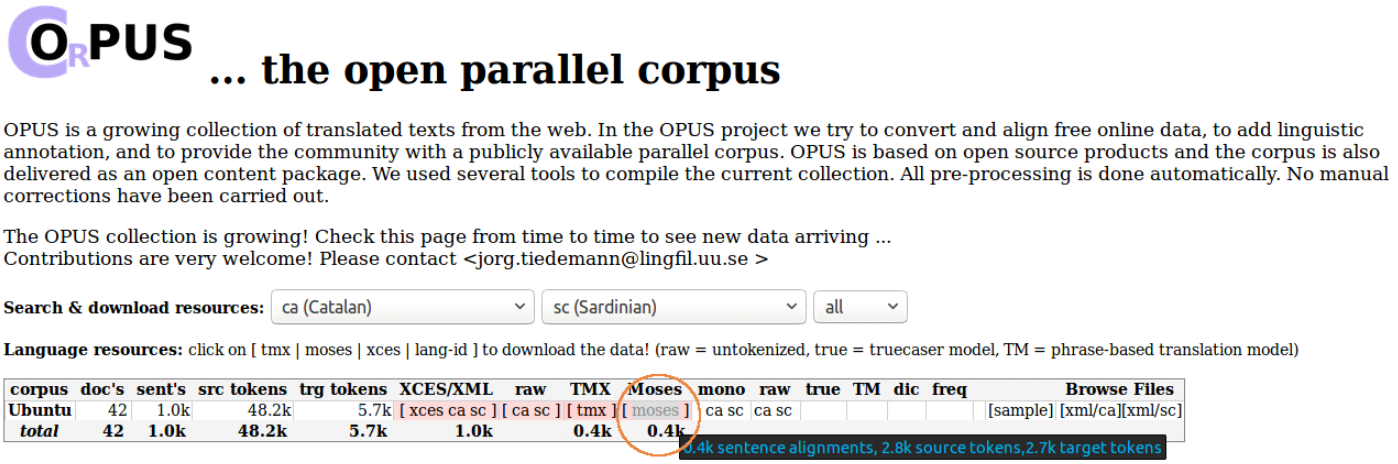

Figure 3 *OPUS website, showing corpora in Moses format*

Future developments should allow uploading bilingual texts, i.e., in TMX format (see 5, *Future developments*), alongside texts in Moses format. The file manager automatically counts the lines, words and characters contained in the uploaded file and also allows you to preview, download and delete texts. The user, therefore, should upload at least two files that contain the same number of lines and which correspond to source and target texts.

The following step is called Monotext manager, where the user can combine two or more files uploaded in the previous phase, provided that they are all written in the same language. These monotexts will be subsequently used to train LMs (step 3, LMs).

The Bitext manager (step 4), in turn, prompts the user to create bilingual corpora by aligning source and target files uploaded in the first step. Users can add as many source–target files as they want, provided that they are parallel (and, therefore, have the same number of lines). These bitexts will be used in the following step (Translator trainer, step 5) to train SMT engines, together with the LM created in step 3 for the target language of the engine.

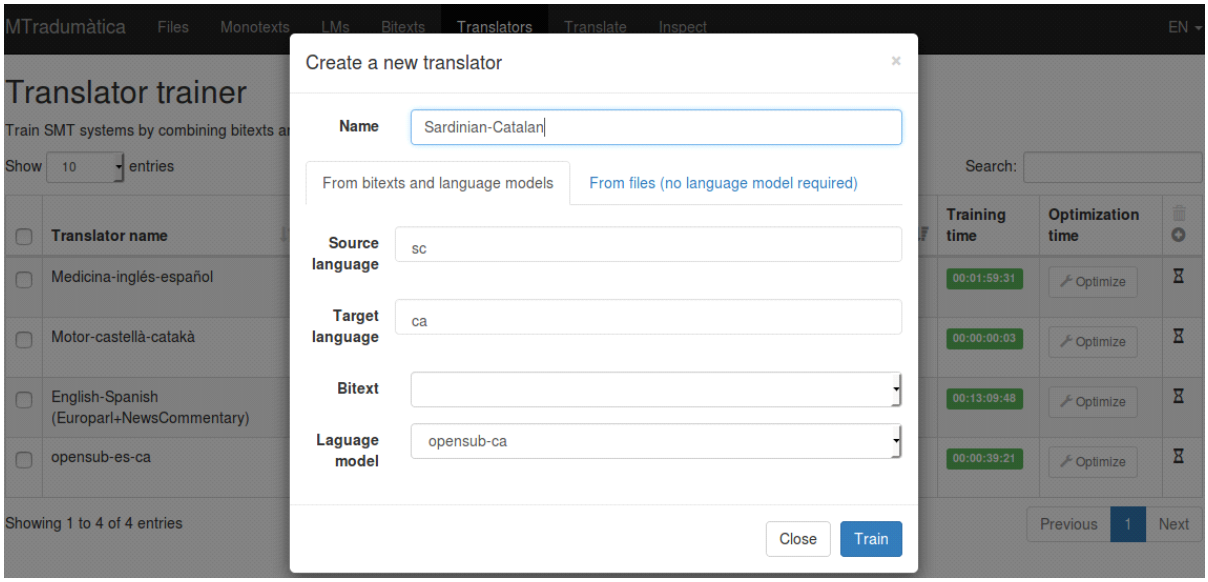

Figure 4 *Step 5:* Create a new translator *dialogue*

Step 5 includes the option of creating a translator from the files uploaded in step 1, without having previously trained an LM. Although the quality of a translator without an LM is likely to be inferior compared to a translator with an LM, it was deemed appropriate to include this option for pedagogical purposes. The Optimize button, visible in figure 4, allows the user to tune the system, but as stated in section 5 (*Description of MTradumàtica: Interface and processes*), this is a very slow process.

The final step (Translate) allows the user to select one of the translators created during the process and to test it by either typing text in the text box or uploading documents. The following file formats are currently supported: HTML, TXT, DOCX, PPTX, XLSX, ODT, ODS and ODP.

At the time of writing this article (December 2016), an additional step (7) called *Inspect* was available on the website. This feature, partially functioning at the moment for demonstration purposes, should allow the user to query and examine the components of the engines already created, i.e., the TM (not yet implemented) and the LM (partially implemented). By entering tokenised text in the Input text box the user will be able to analyse the components of the translator.

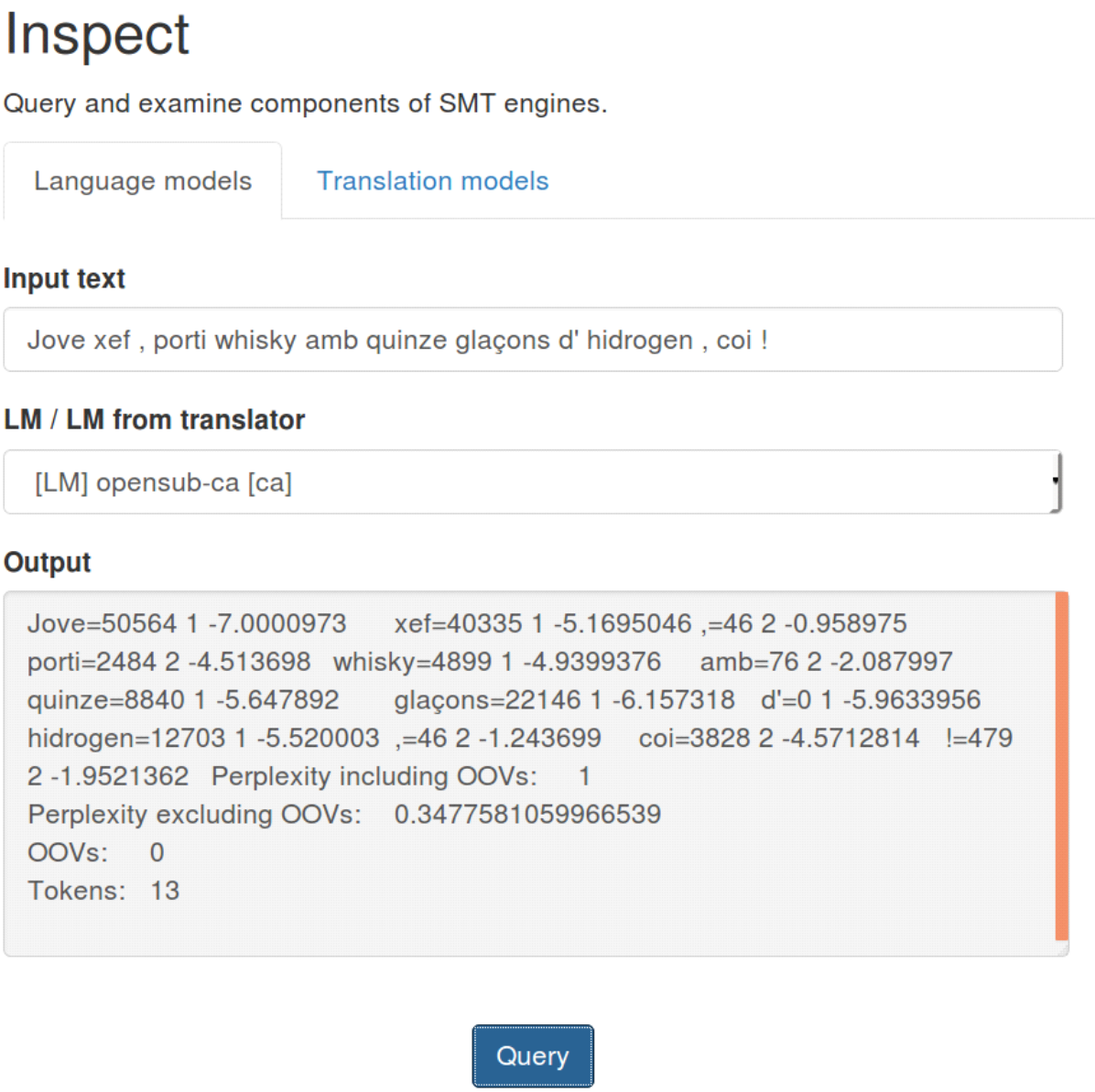

Figure 5 *Step 7:* Inspect *interface showing the analysis of a sentence in the LM*

Words not found in corpora (out-of-vocabulary words, OOVs) and the number of tokens are shown alongside identifiers and statistical measures for each token.[7] Future developments should provide a more accurate analysis of the source text and a similar feature to inspect the translation model (including probabilities from the phrase table).

6. Future developments

MTradumàtica is already available for testing purposes; however, the goal of ProjecTA is to enable translators and small translation enterprises to use it to create their own private engines. It still lacks some functionalities to be useful for them, which should be overcome with future developments. Of these, the following have already been mentioned in the previous sections:

- Uploading TMX files. MTradumàtica currently only accepts corpora in Moses format,

i.e., one sentence per line. This is a rather uncommon format among translators and translation companies, who are much more used to the standard format for translation memories, TMX, given that virtually all translation software allows importing and exporting this file format.

- TM inspect. The previous section has shown that the LMs can already be inspected (available only for testing purposes). There are plans to add the TM inspection in the near future.

Other features have already been discussed, and the following priorities have been identified:

- User management. The implementation of a user management system, including user accounts with username and password, is essential to provide service to users so that they can upload their own corpora and keep their systems private.

- Concatenating and prioritising models through GUI. Whenever various statistical models are available, a GUI should allow users to concatenate and prioritise them through a GUI. For instance, users might want to feed the system with a small TMX of their own and prioritise it so that it overrides probabilities coming from the rest of the corpora. This is closely related to the following feature.

- Terminology management. Users should be allowed to upload glossaries or terminology databases, for instance in CSV (comma-separated values) or TBX (termbase exchange) format —the standard format for terminology databases. The user glossary should override probabilities from the TM, so that user terminology is always *preferred* by the system.

- Integrated corpora management. MTradumàtica should be able to allow users to examine and modify text data contained in corpora from within the application. Thus, users could adjust the quality of the output by editing the corpora. This is closely related to the following feature.

- Automated pre and post-editing functionalities. MT engines are not modified as a result of post-editing actions. Translators, as mere end-users of MT, might feel exasperated with the perspective of having to correct recurrent errors. The correction of recurrent errors previous to the use of the system (e.g., the modification of the engine itself, not of the translation) could be approached from a double perspective.

- Automated pre-editing: On the one hand, translators might want to batch recurrent textual patterns in the original corpora. A search and replace feature with regular expressions would allow users to improve the systems semi-automatically.

- Post-editing interface. On the other hand, the system should provide a post-editing interface in order for the user to process the raw output of the MT system. This interface should allow, as in pre-editing, to batch replace textual patterns in the target-side as well (automated post-editing). More important, however, would be that the system could automatically recognise recurrent post-editing actions by the translator by applying machine-learning techniques. A basic illustrative example would be that, whenever translators repeatedly post-edit a misspelled word, the system prompts them with a dialogue asking whether the same correction should be applied to matches in the corpora. This batch replacing could potentially generate new undesired errors. To

minimise this, a preview dialogue would show concordances in the corpora, so that users could check the context and decide whether the correction should be implemented. In this sense, previous research in the field of post-editing, as mentioned in section 3, could be useful.

- Automatic quality evaluation metrics. These metrics, such as BLEU (see section 4, *Statistical Machine Translation Customisation*) provide an evaluation of the quality of MT. They can also be useful to provide insights into the relevance of the actions carried out while training the system, e.g., whether editing the corpora has any significant effect on the output.

- Integration with computer assisted translation (CAT) tools through APIs. APIs should be generated for third-party software to allow MTradumàtica connect with it. This would allow translators to integrate MTradumàtica with their preferred software so that project managers, for instance, could prepare the engines their translators would be working with. Eventually, MTradumàtica could even be used by users to query strings in corpora (as a concordance feature) or to repair fuzzy-matches from translation memories (Ortega, Sánchez-Martínez & Forcada, 2014).

7. Concluding remarks

This article has described the results of ProjecTA, a project which attempts to bring MT within closer reach of translators, based on the premise that the translator's profile is appropriate for the management of MT-related tasks. The main outcome of the project is MTradumàtica, software specifically designed for SMTC by translators. Hopefully, MTradumàtica —together with complementary training in MT— will allow translators to include MT in their services catalogue through the implementation of their own MT engines.

The present project can be regarded as an effort from within the translation field to contribute in one way to empowering translators to use MT. This translates as independence from third-party systems and, therefore, an increased confidentiality as a result of decentralised installations, i.e., the possibility of installing MTradumàtica on private servers. This empowerment might have, in turn, an effect on the way translators regard MT and how they relate to it: MT might cease to be perceived by translators as a mere end-product providing unalterable results, and translators might cease to think of themselves as mere end-users of MT.

Our research also suggests that the success of using MT relates not only to the quality of the raw MT output (which mainly depends on the corpora owned by the translator), but also to the degree of usability of the systems. MTradumàtica attempts to ease integration with the workflow and to remove most of the technical barriers for the integration of MT in enterprises so that freelance translators and small enterprises can use it.

The experimental version of MTradumàtica is already available at m.tradumatica.net.

Notes:

1 Reference: FFI2013-46041-R, funded by Ministerio de Economía y Competitividad del Gobierno de España. Programa Estatal de Investigación, Desarrollo e Innovación Orientada a los Retos de la Sociedad.

2 While empirical evidence is needed, in the light of recent advances in MT research, NMT has allegedly allowed for significant improvements to the quality of the raw output of some combinations. See https://www.blog.google/products/translate/found-translation-more-accurate-fluent-sentences-google-translate/.

3 See https://www.gnu.org/copyleft/lesser.html.

4 Some packaged versions of Moses do have a GUI, but these are not part of the original Moses project. See http://www.statmt.org/moses/?n=moses.packages.

5 For a glossary on the terminology used by Moses, refer to http://www.statmt.org/moses/?n=Moses.Glossary.

6 See http://opus.lingfil.uu.se/.

7 This function is not producing real figures at the time of writing this article.

References:

Aziz, Wilker; Castilho, Sheila, & Specia, Lucia. 2012. PET: a Tool for Post-editing and Assessing Machine Translation. In *LREC* (pp. 3982–3987). Retrieved from http://wilkeraziz.github.io/dcs-site/publications/2012/AZIZ+LREC2012.pdf, accessed December 17, 2016.

Forcada, Mikel L. 2009. Apertium: Traducció automàtica de codi obert per a les llengües romàniques. *Linguamática* 1(1): 13–23.

Forcada, Mikel L. *et al*. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, *25*(2), 127–144. https://doi.org/10.1007/s10590-011-9090-0

Koehn, Philipp. 2016. Moses User Manual and Code Guide. http://www.statmt.org/moses/manual/manual.pdf, accessed December 17, 2016.

LT-Innovate. 2013. LT2013. Status and Potential of the European Language Technology Markets. http://ec.europa.eu/information_society/newsroom/cf/dae/document.cfm?doc_id=4267, accessed December 17, 2016.

Machado, Maria José; Leal Fontes, Hilário. 2014. Moses for Mere Mortals. Tutorial. A machine translation chain for the real world. https://github.com/jladcr/Moses-for-Mere-Mortals/blob/master/Tutorial.pdf

Ortega, John E.; Sánchez-Martínez, Felipe, & Forcada, Mikel L. 2014. Using any machine translation source for fuzzy-match repair in a computer-aided translation setting. In *Proceedings of the 11th Biennial Conference of the Association for Machine Translation in the Americas* (Vol. 1, pp. 42–53). Retrieved from http://www.dlsi.ua.es/~mlf/docum/ortega14p.pdf, accessed December 17, 2016.

Papineni, Kishore; Roukos, Salim; Ward, Todd, & Zhu, Wei-Jing. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 311–318). Association for Computational Linguistics. Retrieved

from http://dl.acm.org/citation.cfm?id=1073135, accessed December 17, 2016.

Ping, Ke. 2009. Machine translation. In Mona Baker & Gabriela Saldanha (Eds.), *Routledge encyclopedia of translation studies* (Vol. 2, pp. 162–168). London: Routledge.

Torres-Hostench, Olga; Presas, Marisa, & Cid-Leal, Pilar (coords.). 2016. *L'ús de traducció automàtica i postedició a les empreses de serveis lingüístics de l'Estat espanyol. Informe de recerca ProjecTA 2015*. Bellaterra. Retrieved from https://ddd.uab.cat/record/166753, accessed December 17, 2016.

Adrià Martín-Mor
K1020
Campus UAB
08193 Bellaterra (Barcelona)
Catalonia
e-mail: adria.martin@uab.cat
ORCID: 0000-0003-0842-3190

In SKASE Journal of Translation and Interpretation [online]. 2017, vol. 10, no. 1 [cit. 2017-28-04]. Available online <http://www.skase.sk/Volumes/JTI12/pdf_doc/02.pdf>. ISSN 1336-7811