On the Need of New Tools for "Translating Writers" in Industry

Claire Lemaire, Christian Boitet

On the Need of New Tools for "Translating Writers" in Industry

Claire Lemaire
Laboratoire d'Informatique de Grenoble
claire.lemaire@imag.fr

Christian Boitet
Laboratoire d'Informatique de Grenoble
christian.boitet@imag.fr

Abstract

Working in the context of French and German companies, we discovered the emergence of a new situation of *bilingual writing*, where French or German technical writers writing in their (source) language (SL) are asked to produce a parallel version of their document in English (the TL), often for delocalization purposes. These *technical translating writers* cannot benefit from available tools such as MT+PE (post-editing environment) or TM-based translator aids to produce good enough translations. But, not only in IT, badly translated requirements and specifications lead to the development of totally inadequate products. We propose a scenario using existing free tools such as MT, sub-sentential aligners, company-specific bilingual terminology, SL and TL correctors, and integrating iterations of (re)writing their SL text, MT-translating it, correcting it somewhat, and translating it back from TL to SL. We then outline a more futuristic approach, relying on a multiple SL analyzer, an interactive disambiguator, the production of a "self-explaining" document (SED), and the subsequent automatic generation of a high-quality TL document in SED format. In short, the aim would be to build a true *bilingual writing tool for technical translating writers.*

1 Introduction

Working in the context of some French and one German company, the first author discovered the emergence of a new situation of *bilingual writing*, where French or German technical writers writing in French or German, their native tongue (the source language, or SL) are asked to produce a parallel version of their documents in English (the target language, or TL), often for delocalization purposes (Lemaire, 2016). These *technical translating writers* know some English (say, to a B1 level) and have often access to the specific bilingual terminology used in their technical context. As their management ignores the requirements for producing good translations, they get no support, and they end up using free tools like Google Translate or Bing (Lemaire, 2017). As no revision (and even no proof-reading) is done on the results, the quality is very bad, with sometimes disastrous consequences.

In fact, if their company does not want to pay professional translators to do a decent job, it seems there is nothing they can do to solve this problem.

- They can't buy a cheap license for a "good enough" MT system and let the results be post-edited by the technical translating writers: (1) specialized MT systems may be very good, but are somewhat expensive, as they must be built from good translation memories (TMs) and specific bilingual term banks (TBs), and (2) in any case, post-editing into English to get high quality translations can only be performed by native speakers of English knowing the domain well.

- They also can't train their technical translating writers to use TM-based like SDL-Trados, although some recent ones are free (OmegaT[1], SmartCAT[2], Poedit[3], MateCat[4]

[1] http://sourceforge.net/projects/omegat
[2] https://www.smartcat.ai/
[3] https://poedit.net/
[4] https://www.matecat.com/

Proceedings of the 39th Conference Translating and the Computer, pages 70–75,
London, UK, November 16-17, 2017.

and many others[5]), because they are all tailored to professional translators translating into their native tongue.

We think that, despite that apparent impossibility, it should be possible to help technical translating writers produce translations of reasonable quality using only existing free tools such as MT, sub-sentential aligners, company-specific bilingual terminology, and SL and TL correctors. What would change is the scenario of the production of *both* the SL and TL documents. Instead of SL document writing → MT → TL (LQ = low quality) document, we would introduce a loop of the form, at the (n+1)-th iteration:

SL_doc1_{n+1} (re)writing →MT →TL_doc1_{n+1} → checking →TL_doc2_{n+1} →MT→SL_doc2_{n+1}.

Note that this approach would not be usable in a classical translation context, because translators must start from the SL text as it is. They can correct typos in passing, but they are not allowed and even less asked to modify it. But our technical translating writers are the *authors* and can therefore write and rewrite until the translation seems them (aided by the correctors, the aligner and the reverse MT) to be grammatically and terminologically correct.

This first approach will be detailed in section 3.

The second approach we propose is much more futuristic, although it builds on ideas that have been successfully prototyped in the past, in particular at IBM-Japan (JETS system). It relies on the (demonstrated) possibility to build an interactive SL disambiguator coupled with an "all-path" parser and a bilingual dictionary aligning SL and TL lexemes and word senses.

In line with the "semantic Web", it also introduces the idea to add to a SL document annotations contained in a *companion* document and comprising everything that is needed to show the ambiguities (relative to the SL→TL pair or the SL→TL1/TL2…/TLn pairs), how they have been solved. It has been shown (back in 1994!)[6] than a SL SED text can be translated totally automatically in a corresponding TL SED text. In short, the aim of that more futuristic approach is to build a true *bilingual writing tool for technical translating writers.*

2 More on the business situation

Many companies need to produce enormous quantities of documents in many languages with a very high quality. Also, the terminology of software products is specific to the company. When Bull sold IBM AS-4000 workstations under AIX[7] in OEM, it translated the AIX documentation in its own "Bull-AIX-French" and did not use the "IBM-AIX-French" existing translations.

Almost all companies outsource translations to translation agencies or to freelancers. A big problem is that the cost is high (counting everything, about 0.15€/word for en→fr or fr→en, often more for more distant pairs, for smaller markets). Another is that the number of target languages has increased and is increasing. Commercial companies like Microsoft, IBM or Adobe translate their products (external documentation, on-line help, interface elements like button and window labels, menu items, system messages) in 40 to 60 languages[8].

For that, they use professional translators and propose or require them to use specific tools and resources, like the TM™ tool, a MT system, and, for each translation job, a kit containing a specific bilingual terminology, and a document-dependent TM (extracted from the enormous main TM, and more practical and useful on a PC). The annual size of translated

[5] http://termcoord.eu/2016/06/139-free-tools-suggested-by-professional-translators/

[6] by Boitet & Blanchon at MT25YON, at Cranfield, in 1994.

[7] IBM proprietary version of Unix.

[8] https://console.bluemix.net/docs/services/language-translator/index.html#supported-languages lists 62 languages. Mozilla, a non-profit open source collaborative project, localizes its tools to at least 116 languages (see https://addons.mozilla.org/fr/firefox/language-tools/). Office 2016 has 39 language packs (see https://www.itechtics.com/download-free-office-2016-language-packs-languages/).

documents is often over 20M words per year (10 years of EuroParl!). That is the same for service companies like SAP.

It is then understandable that these companies try to diminish the cost of translation for the "grey" (internal) part of what they have to translate. That is why, for example, SAP stopped doing the translation of requirements and specifications documents in a professional way and asked their writers to become "technical translating writers". In one case we learned about (not at SAP), raw MT translations of functional specifications were sent to a development team based in India. The French client complained that the product did not meet its specifications, and menaced to sue the company, who then had to send an experienced engineer-developer on site for 3 months to develop a correct product. After all, bad translation can end up costing much more than professional translation!

One should remark here that this change of practice may well have been caused by the profusion of loud claims made by MT developers concerning the increase in quality of MT systems, to the point that some are claiming that NMT (neural MT) systems are now as good or even better than professional translators. That is in general utterly false, and can be true *only* in the case of MT systems (following whatever paradigm, expert or empirical) *specialized to a small enough sublanguage*, such as the METEO system for weather bulletins[9] or the ALTFLASH system for Nikkei flash reports.

Returning to the situation in companies wanting to turn their technical writers into technical translating writers, which would be the (sole) users of an environment meant to help them:

- The technical writers know the terminology very well, in both their language and English.
- There may be some native speakers of English in the company, but probably none or very few doing the same job of technical writing — and then, they might or might not have to produce a version in the "local" language (in our cases, French or German).
- The texts concerned are IT requirements and functional specifications, that is, exclusively technical translations.
- The writers are not "recognized" for producing a parallel version in English: they get no special financial incentive, no feedback from anybody in the company, and usually no feedback either from the delocalized development team, whose members, not native speakers of English, are often not competent enough in English to be sure that the purported "translation" is erroneous or even outright meaningless, so that they try to guess a meaning that could "make sense" in the context — and of course often fail.

3 Approach 1: integrate existing free tools in a new scenario using "rewriting"

In this section, we would like to give some details on our first proposed approach, and show that it should indeed be possible to help technical translating writers produce translations of reasonable quality using only existing free tools such as MT, sub-sentential aligners, company-specific bilingual terminology, and SL and TL correctors. The scenario, introduced in 1 above, is to induce the technical translating writer to (1) correct what s/he can in the MT results, namely the terminology, and (2) rewrite her/his SL text so that the MT-translated version improves.

In this scenario, the technical translating writer produces *both* the SL document and the corresponding TL document in an iterative way. The (n+1)-th iteration would be of the form:

SL_doc1_{n+1} (re)writing $\rightarrow$MT $\rightarrow TL\_doc1_{n+1} \rightarrow$ checking $\rightarrow TL_doc2_{n+1} \rightarrow$MT$\rightarrow SL_doc2_{n+1}$.

The first step, writing (if n=0) or rewriting (if n>0) contains the use of some classical tools, such as a spell-checker and a grammar checker. For the second step, MT, we would use whatever MT system the technical translating writer already uses. An improvement here could

[9] And not for the whole domain of weather forecasting, that also contains situations and warnings.

be easily introduced. It would be to use *several* (2 or 3) free MT systems and to select the result having the best score according to some quality estimator (QE). Research on QE has already produced convincing results.

Then, the MT result, TL_doc1$_{n+1}$, would be checked in 2 ways. (1) A language checker would signal spelling and grammar errors. The user is supposed to have at least a B1 level in English, which is normally enough to understand the error, if any, and to accept or not the proposed correction. (2) An aligner such as Giza++ or Anymalign (Lardilleux, 2010) combined with the bilingual term bank would show the correspondences between segments, and in particular between source and target terms, colouring them (for example) in green if they are in the term bank, and in red if they are not.

The resulting TL document, TL_doc2$_{n+1}$, would then be "back-translated" into the SL, producing SL_doc2$_{n+1}$. On that basis, the writer could perceive whether the translation still has problems or not. If yes, the writer would enter the next iteration. S/he would modify SL_doc1$_{n+1}$ to produce SL_doc1$_{n+2}$, the input to the (n+2)-th iteration.

This new kind of help might be implemented as a web service, residing in a server of the company. It should probably allow the user to perform the iterations sentence by sentence, or paragraph by paragraph, or on the whole text. It would be interested to see which strategy would be preferred by the technical translating writers, and which would give the best results.

4 Approach 2: towards a true bilingual writing tool for technical translating writers

Our second solution could be developed in 2-3 years in a particular context, then generalized. It builds on ideas successfully prototyped in the past, in particular at IBM-Japan (JETS system), and then by our LIDIA project (1990-1995), on which H. Blanchon did his PhD, and which was an essential part of the Eureka EuroLang project (Boitet & Blanchon, 1994). In short, the idea is to build an interactive SL disambiguator coupled with an "all-path" parser and a bilingual dictionary aligning SL and TL lexemes and word senses.

Typically, after the writer has written a paragraph, s/he clicks on it to say that its segments (usually sentences) can be processed. The segments are sent to a web service that returns, for each segment, a factorizing "mmc-structure" containing all linguistically and especially lexically possible representations. In the example below, that structure is a tree containing 2 subtrees, one for each representation for the sentence: "Which author cites this speaker?".

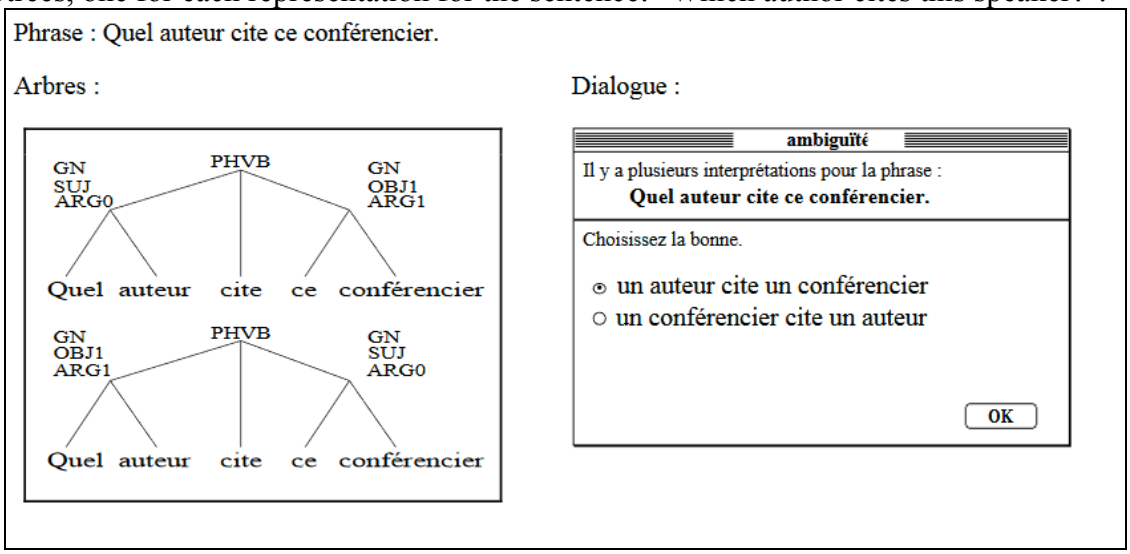

Figure 1 : Interactive functional (subject/object) disambiguation, Blanchon (1992)

Another module then identifies all ambiguities and generates a "question tree", according to a certain strategy, for example ordering them by type, or by cruciality. Both the mmc-

structure[10] and the question tree are then returned to the writing environment. A button appears next to the (SL) segment to signal to the writer that the system is ready to ask questions. It is very important here that the writer is not obliged to answer immediately, and can continue whatever task s/he is engaged in. A human should never be slave of a machine!

If and when s/he feels like it, the user clicks on the dialogue button and answers the questions. Contrary to the LIDIA prototype and to all interactive translation systems we know of, this new system should allow the user to leave the disambiguation dialogue at any point, leaving to the following modules the task of handling the remaining ambiguities automatically, either by selecting for each a solution with the best score, or by producing a factorized output, like, in en-fr, "plante/usine/espion" (plant/factory/mole) for "plant".

At that point, the system will add to the SL segment annotations contained in a *companion* structure and comprising everything that is needed to show the ambiguities, and how they have been solved, that is, the mmc-structure, the question tree, and the answer to each question down the "disambiguating branch".

The resulting umc-structure would then be sent to a web service performing the remaining steps of translation, using a transfer or abstract pivot architecture. That is not important for our user. What is important is that, because the representation is unambiguous, a classical generation process, once debugged and tuned properly, will produce very HQ results.

Nevertheless, ambiguities will almost certainly appear in translations. To eliminate them, the idea is to parse the TL text with an all-path parser built as the inverse of the generator, giving rise to a mmc-structure that contains the umc-structure produced as an intermediate step during generation. It will then be possible to run the interactive disambiguator of the TL automatically: a program will replace the human, and, for each question, select the answer that itself selects the subset of the current set of structures that contains the goal (the starting umc-structure). Hence, the TL segment will be representable in a SED format.

That approach would certainly enable technical translating writers to produce very precise, grammatical, and semantically exact English versions. Nevertheless, we should keep the possibility of correction in the TL, for the terminological part, and leave open the possibility of rewriting the SL text, at least for an interesting reason: it sometimes happens that some information from the context is not explicit in the SL text, but should imperatively be explicit in the TL text. That situation is very common when translating from Japanese into English or French or German, but it also happens between near languages, such as English and French: "he was" → "il avait 1 an" / "il faisait 1 m" →"he was 1 year old" / "he was 1 meter tall". In such cases, the solution would probably be to rephrase the SL text in a more explicit way.

5 Conclusion & perspective

We are embarking on an internal project to implement and evaluate our first approach, and are looking for a company that would like to experiment it with us.

Concerning the second approach, it is a longer-term project, which we have begun to work on with CS (Communications and Systems) in the framework of a project preparation. Here, the domain would be the writing of system requirements, representing them after disambiguation as UNL[11] graphs, then as UML graphs, and further as logical expressions in the specific domain ontology (Sérasset & Boitet, 2000). Starting from any of the last 3 forms, one would be able to generate the requirements in several languages and forms, in particular SED forms and controlled language forms.

---

[10] mmc: multiple, multilevel and concrete; umc: unique, multilevel, concrete; uma: unique, multilevel, abstract.

[11] UNL (Universal Networking Language) is a language of « anglo-semantic » hypergraphs able to represent any utterance in any natural language. Arcs bear semantic relations and nodes bear interlingual lexemes (UWs) and semantic features. See http://undl.org.

Acknowledgments

Our first thanks go to the ANRT (Agence Nationale de la Recherche Technologique) and to the L&M (Lingua et Machina) company, that have supported the first author for 3 years. We are also very grateful to the firms that have allowed us to look into their translation practice: SAP, Vicat, and EDF while we worked with L&M.

References

Blanchon, Hervé. 2004. Comment définir, mesurer et améliorer la qualité, l'utilisabilité et l'utilité des systèmes de TAO de l'écrit et de l'oral. Une bataille contre le bruit, l'ambiguïté, et le manque de contexte. *Mémoire de HDR*, 380 p., Université Joseph Fourier, Grenoble.

Blanchon, Hervé. 1992. A Solution to the Problem of Interactive Disambiguation. *COLING 1992,* 23–28 juillet. Nantes, France. 1233–1238.

Blanchon, Hervé. 1994. LIDIA-1 : une première maquette vers la TA Interactive "pour tous". *Thèse de doctorat,* Université Joseph Fourier – Grenoble I, Grenoble, 1994.

Blanchon, Hervé, and Boitet, Christian. 2007. Pour l'évaluation externe des systèmes de TA par des méthodes fondées sur la tâche. TAL, 2007: 33–65.

Boitet, Christian. 1995. Factors for success (and failure) in Machine Translation–some lessons of the first 50 years of R&D. *Proc. MTS-V (Fifth Machine Translation Summit)*, 11–13 juillet, Luxembourg.

Boitet, Christian. 1976. Un essai de réponse à quelques questions théoriques et pratiques liées à la traduction automatique : définition d'un système prototype. Modélisation et simulation. *Thèse de Doctorat d'État*, Université Scientifique et Médicale de Grenoble, 250 p.

Boitet, Christian, and Blanchon, Hervé. 1994a. Promesses et problèmes de la "TAO pour tous". Après LIDIA-1, une première maquette. *Langages. Le traducteur et l'ordinateur*, sous la direction de Jean-René Ladmiral, 1994a: 20–47.

Boitet, Christian, and Blanchon, Hervé. 1994b. Multilingual Dialogue-Based MT for monolingual authors: the LIDIA project and a first mockup. *Machine Translation 2*, 99–132.

Chan, Andy Lung Jan. 2010. Perceived benefits of translator certification to stakeholders in the translation profession: A survey of vendor managers. *Across Languages and Cultures 11*, n° 1, 93–113.

Dam, Helle, and Zethsen, Karen. 2010. Translator status: Helpers and opponents in the ongoing battle of an emerging profession. Target 22, n° 2, 194–211.

Huynh, Cong Phap, Valérie Bellynck, Christian Boitet, et Hong Thai Nguyen. 2010. The iMAG concept: multilingual access gateway to an elected Web sites with incremental quality increase through collaborative post-edition of MT pretranslations. *TALN-2010*, Montréal.

Huynh, Cong-Phap. 2010. Des suites de test pour la TA à un système d'exploitation de corpus alignés de documents et métadocuments multilingues, multiannotés et multimédia. *Thèse de doctorat,* Université Joseph Fourier.

Lardilleux, Adrien. 2010. Contribution des basses fréquences à l'alignement sous-phrastique multilingue : une approche différentielle. *Thèse de doctorat,* Université de Caen - Human-Computer Interaction.

Lemaire, Claire. 2016. Linguistic methodology to help German and French non-translator users to write bilingual specifications. *38. Jahrestagung der Deutschen Gesellschaft für Sprachwissenschaft*. Universität Konstanz.

Lemaire, Claire. 2017. Traductologie et traduction outillée : du traducteur spécialisé professionnel à l'expert métier en entreprise. *Thèse de doctorat*, Université Grenoble Alpes.

Nguyen, Hong-Thai. 2009. Des systèmes de TA homogènes aux systèmes de TAO hétérogènes. Interface homme-machine. Thèse de doctorat, Université Joseph-Fourier – Grenoble I.

Sérasset, Gilles & Boitet, Christian. 2000. On UNL as the future "html of the linguistic content" & the reuse of existing NLP components in UNL-related applications with the example of a UNL-French deconverter, Proc. *COLING 2000*, Saarbrücken, Germany, 7 p.

Slocum, Jonathan. 1985. A survey of machine translation: its history, current status, and future prospects. *Computational linguistics* 11.1, 1–17.

Vasconcellos, Muriel. 1995. Advanced software applications in Japan. Elsevier.

Vasconcellos, Muriel. 1993. The Present State of Machine Translation Usage Technology; Or: How Do I Use Thee? Let Me Count the Ways. *MT Summit IV*, July 20–22. Kobe, Japan. 47–62.

Zhang, Ying. 2016. Modèles et outils pour des bases lexicales "métier" multilingues et contributives de grande taille, utilisables tant en traduction automatique et automatisée que pour des services dictionnairiques variés. Thèse de doctorat, Université Grenoble Alpes.