PangeaMT - putting open standards to work... well

E. Yuste & M. Herranz

Pangeanic (PangeaMT)

Trade Center, Prof. Beltrán Báguena 4
E – 46009 Valencia, Spain
eyuste@pangea.com.MT

A-L. Lagarda, L. Tarazón, I. Sánchez-Cortina, & F. Casacuberta
Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
Camino de Vera, s/n
E – 46009 Valencia, Spain
alagarda@iti.upv.es

Abstract

PangeaMT is presented from our standpoint as a LSP keen to develop and implement a cost-effective translation automation strategy that is also in line with our full commitment to open standards. Moses lies at the very core of PangeaMT but we have built several pre-/post-processing modules around it, from word reordering to inline mark-up parser to TMX/XLIFF filters. These represent interesting breakthroughs in real-world, customized SMT applications.

1 Introduction

Pangeanic[1] is a Spanish language service provider (LSP) that works for global and highly specialized enterprise clients, other LSPs, and cross-national institutions. In mid 2000s, operational workflows based on CAT tools just proved insufficient and even inefficient at times. There was an urge to reduce turn-around times (TAT) as there was a higher demand of translation work. Increasing translation productivity through translation automation and MT output post-editing was to be the answer. But what MT system could really be of use? Our long-term clients posed us with the same question. We all were in need of fully-tailored, domain-specific MT solutions that helped us reduce costs and not impose yet another dear piece of software.

In terms of customized development, we could afford neither the time nor the expense to add hundreds of syntax or lexical rules to existing systems (some of them conceived ages ago).

After evaluating commercially available MT systems and learning deeply about MT, moving forward to Statistical Machine Translation (SMT)[2] development and consulting simply had to happen. In 2009 Pangematic, our self-enhanced MT system based on Moses[3] was born. Soon after, Pangematic would be renamed as PangeaMT[4].

Section 2 of this paper presents PangeaMT engines as domain-specific and customized SMT solutions with open-standard capabilities. In order to understand this *openness* paradigm, the system's components and the overall open-standard geared philosophy are discussed. Section 3 then focuses on PangeaMT in use and backs up the adoption of such engines in either the internal workflow of an LSP or a corporation by looking at promising real-life data, both objective and subjective, resulting from deployment-related figures and user/client appreciations. We finally draw some conclusions and discuss ongoing and future work.

2 PangeaMT – Pangeanic's domain-specific, customized and stats-driven MT solution range

As already pinpointed in section 1, PangeaMT is an evolution of Moses. Engine development is meant to be quicker under the SMT framework.

SMT works much better for specific language domains, which is what we and our clients needed.

[1] http://www.pangeanic.com

[2] Statistical machine translation engines work by automatically learning which words / set of words, are translated for which, also taking into consideration which context they occur in.

[3] http://www.statmt.org/moses/

[4] http://www.pangea.com.MT

Once you have created an engine that works well[5] for a certain language domain, building a new engine can be achieved pretty successfully by adding new pairs of sentences translated from a similar domain / linguistic style.

Realistically speaking, domain-specific build time from scratch in a new language pair will still take three to four months. Depending on client and domain data availability and status, domain type and languages being handled, this may vary.

Some clients get really involved in the process of data gathering, but what if their bi-text collection is far too small? As data consultants, in those situations we normally resort to the Translation Automation User Society - TAUS[6] Data (TDA)[7], an open-standard geared, bilingual data repository created by professionals with the right language industry know-how. Being acquainted with data mining and alignment techniques may also be of help, and so is a willingness to ensure that the data for the engine training corpus is clean and representative.

SMT training and development, testing and implementation represent the core of the process, and perhaps the one clients get least involved with. Our programmers will make use of the data provided, self-generate content from it to build larger and specific content, build language models and, finally train the engines. This constitutes the basis for *engine customization*.

However, customized engines can only be of use if they can be part of and serve open standards workflows, i.e. they can process and generate inputs and outputs in GILT[8] industry standard formats, such as TMX[9] or XLIFF[10]. We will describe (the implications of) this later on.

[5] It works! - unlike other approaches, whereby one has implemented all the linguistic rules and patches one could think of, and it is not yet clear how to adapt to a similar linguistic domain.

[6] http://www.translationautomation.com/

[7] http://www.tausdata.org/

[8] Acronym of Globalization / Internationalization / Localization / Translation. It sums up the core work and expertise areas of the so-called language industries. In the last decades we have seen a significant need of and growing interest in automation, not only in terms of workflow optimization but also machine translation.

[9] http://www.lisa.org/Translation-Memory-e.34.0.html

[10] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xliff

2.1 The stats-driven MT approach that goes far beyond the plain text I/O

PangeaMT has proprietary peripheral modules at the pre-processing[11] and post-processing[12] stages, which allow the system to process files to translate not just in plain text but in other industry open formats.

What follows is a description of the PangeaMT main (M) and peripheral (P) modules, which can also be identified in Figure 1 below:

- **TMX Parser** (P): Reads TMX files and parses its content, extracting sentences in plain text. Developed in *Python* using the *xml.sax* library. This has been followed by the **XLIFF Parser** (P), available in summer 2010.

As a result of these parsers, PangeaMT solutions can read, apart from plain text files, both TMX and XLIFF files. These represent significant technological advances within a stats-driven MT framework. These parsers also own a format generator feature, that is, the system can parse or spot the text to translate and the wrapping TMX/XLIFF tags, then restore those tags, and generate the translated text in its original open standard format.

- **Phrase Coder** (P): Performs pre-processing techniques, mainly on numbers, punctuation marks, parentheses, and other symbols. Developed in *perl*, *bash* and *awk* scripting.
- **Phrase Decoder** (P): Performs post-processing techniques, mainly reversing the Coding process. Developed in *perl*, *bash* and *awk* scripting.
- **Moses Toolkit** (M)
- **IRSTLM Toolkit**[13] (M)
- **Inliner** (P): Estimates placing of inlines in the translated phrase. Developed in *Python*.

This Inliner also represents a breakthrough. SMT solution providers have traditionally focused on producing MT output, normally in plain text

[11] The transformation applied to the source text to train the engine so that this gets translated is called *pre-processing*.

[12] Similarly, the transformations needed to convert the initial output of an MT system to a text that is more useful and understandable by a human user (correct casing, spacing, punctuation, adequate HTML/XML code placing, etc.), are called *post-processing*.

[13] http://sourceforge.net/projects/irstlm/

only. This has not had a positive impact on the overall appreciation of the usefulness of SMT on the part of the corporate users that would like to use machine translation, also for formatted texts, that is, (heavily) codified, marked-up content.

their choice (if they have commissioned different engines in a number of language combinations and knowledge domains) and then one of these options: **Tag-Optimal** and **Trans-Optimal**.

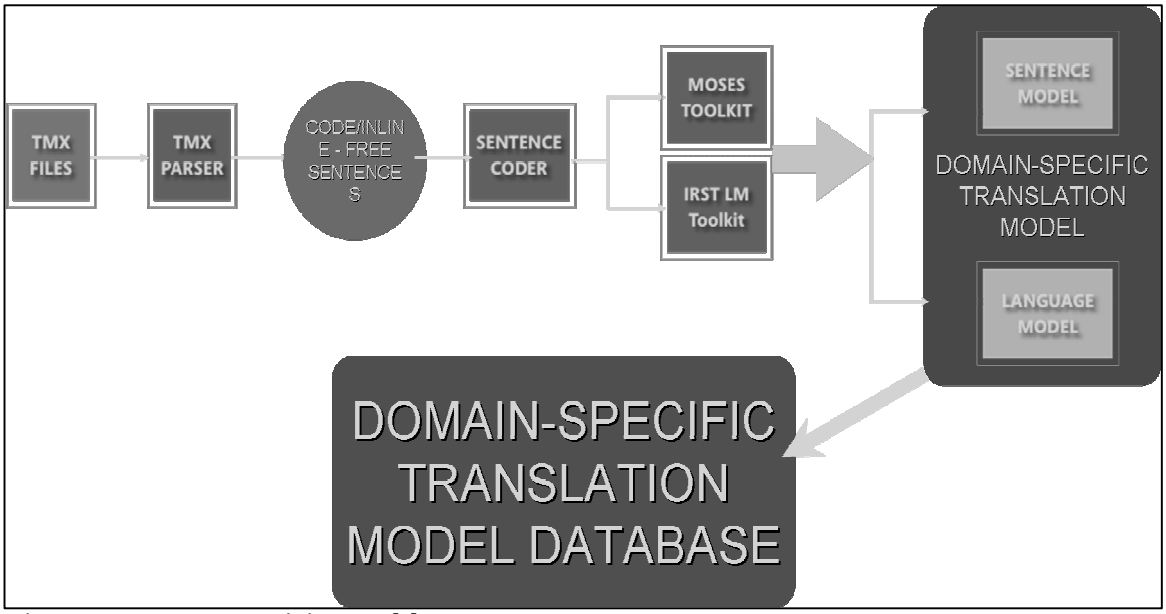

Figure 1. PangeaMT Training Model

Figure 2. Tag-Optimal and Trans-Optimal selection feature

PangeaMT solutions delivered in a web interface have an interesting interactive feature in this respect. In such interface, before uploading the file to translate in a TXT, TMX or XLIFF format, the user can select the language pair and domain of

If the user knows that the file to translate contains lots and lots of inline code, it would be advisable to choose the first option. The Tag-Optimal

function will be called up to ensure that the system concentrates on spotting and then replacing inline code as correctly as possible. This is a challenging task, and therefore the translation quality of the output may suffer a bit.

The Trans-Optimal option[14] should then be selected if the file to translate is not particularly rich in inline code or if translation quality is rather more important. Figure 2 highlights these two options in the PangeaMT web interface.

Once the user clicks on the Translate button, the machine translation process begins. The pre-processing, translation and post-processing components interact and lead to the MT output, as shown in Figure 3.

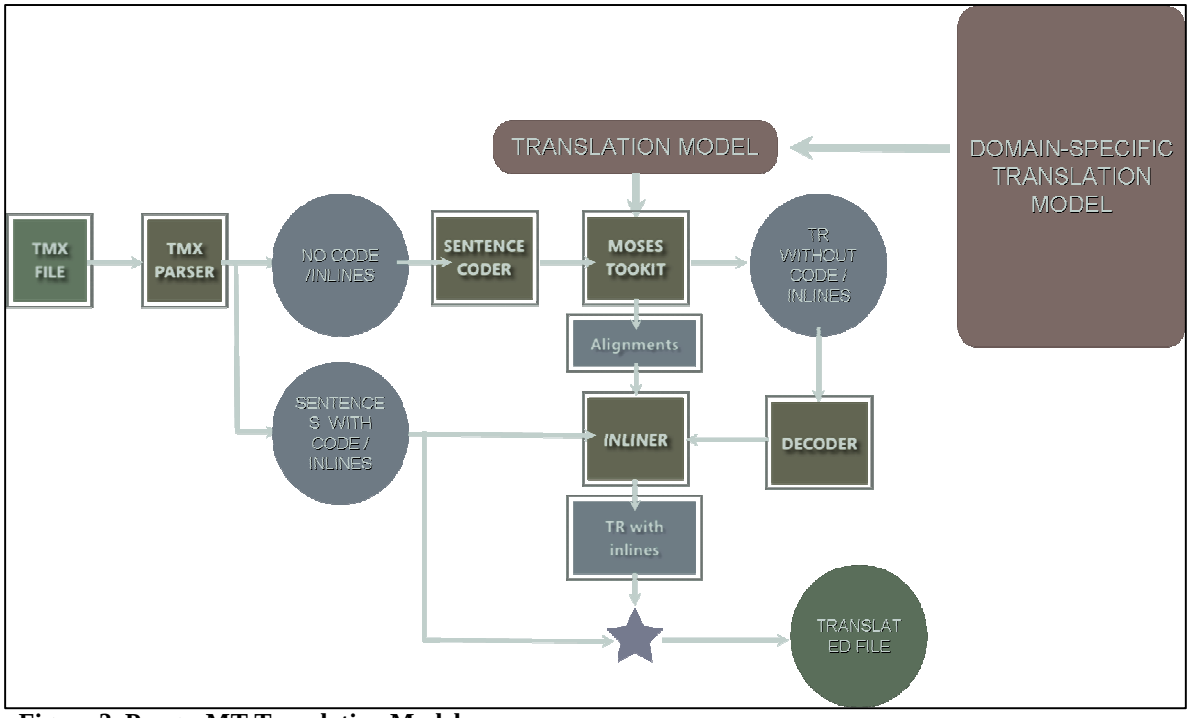

Figure 3. PangeaMT Translation Model

Please note the correlation between figures 1 and 3 here, and look at the modules that are responsible for PangeaMT open-standard treatment capability. In the next subsection, we explain what practical implications these MT technology advances have.

[14] In other stand-alone, customized developments where translation itself mattered the most, even if it had been acknowledged that inline code parsing was a great feature, it was agreed that tags would be placed in a fixed spot in the segment, say at the beginning or at the end. Post-editors would then just have to be aware of this and replace the inline coding accordingly. This is another development option still available upon request.

2.2 Practical implications of using a system conversant with open standards

Deploying a customized MT solution that reads open standards implies technology independence, interoperability, ease of integration.., in one word, freedom to handle, process, and leverage the related content in platforms and programs that also read the same standards.

This freedom is obviously linked to cost-effectiveness[15]. There are no expensive lock-ins, no expensive upgrades with PangeaMT. There will be a need to retrain or update the customized system with the client's post-edited[16] material a few times and at a low cost. Once the solution reaches maturity, there will be little need for maintenance. The client (or the user) can then concentrate on producing more and more translated material.

These practical implications of open-standard-driven PangeaMT engines should be of interest for corporate users and LSPs alike. The next section provides them with further details about the system in use.

[15] Typically, an engine pays for itself in saved translation fees before one year. An engine retraining or update with post-editing material is a fraction of that cost.

[16] If you would like to get to know about the recommended procedure for post-editing (PE), please read Appendix A.

3 PangeaMT in use – internally and at the client's end

This section summarizes two real-world application scenarios of PangeaMT:
1. It exemplifies how the system has been created and is being used internally for machine translation purposes in the automotive domain, one of Pangeanic's main specialization fields;
2. The corporate scenario deployment of PangeaMT is here represented by a summary of the customized solutions for Sony Europe and Sybase.

3.1 From birth to young adulthood of the PangeaMT AUT engine

PangeaMT is now accessible via a web interface or offered as a customization package built anyway for the needs of the following industries:

- Engineering
- Automotive
- Electronics / Computer Hardware and Peripherals
- Telecommunications
- Computer Software
- Marketing-Economy-Business
- Legal-Professional services
- Life Science / Medicine

The list could be refined to reflect developments in some related domains or sub-domains. However, it all had to start somewhere, as explained in the Introduction.

Paying attention to our internal needs first, one of the first domain- and client-specific developments had to target the automotive domain. We have a long-standing big account from a well-known Japanese automotive manufacturer, who also showed a growing tendency toward higher outputs across all kinds of content (in this same order):

- owner's manuals
- technical shop / repair instructions
- fault detection software / UI
- online documentation

We started the corpus cleaning and assembly oriented to SMT engine training from scratch in the automotive (AUT, for short) domain for the ENES language pair in early 2009. The training corpus would amount to 8 million words at the time. The same process for the same domain in the remaining FIGS[17] languages followed shortly.

In early June 2009 the first ENES AUT engine was born. System retrainings (updates) took place in September 2009, January and June 2010, the month from which we extracted the following analytical data[18].

Training	Sentences	1,661,780
	Running words	11,366,260
	Vocabulary size	163,259
Test	Sentences	2,000
	Running words	14,028
	OOVs	88
	Perplexity (3-gram)	16.99

Figure 4. PangeaMT AUT engine training matrix (June 2010)

At that time the system was already above childhood, so to speak, having over 11 million running words. By the time you read this article the next retraining will have already taken place. At the time of writing we estimate that 15 million will at least be reached then.

3.1.1 Monitoring the benefits of the AUT engine deployment

Before starting to use the PangeaMT AUT engine in Pangeanic, a typical automotive translation project of 1 to 2 million words run in a TM+QA-based workflow would take about 6 weeks to complete. Since the system began operation, we have kindly asked our collaborators to monitor and report usage findings that are worth mentioning.

When the automotive engine became more actively used in Pangeanic at the end of 2009, it was acknowledged that production time was shortened by one week to five weeks. The engine would still be retrained with new post-edited material a second time in January.

In Spring 2010, the production time was again shortened by one week to four weeks without sacrificing QA procedures. This remarkable higher

[17] Industry abbreviation of French, Italian, German and Spanish.

[18] Perplexity is a metric from information theory that is useful to evaluate the complexity of a corpus (Rosenfeld, R). Please note that the perplexity found here is really low. This is due to limited cross-textual content variation and to the likely overlapping between the training and the test.

throughput was achieved as a result of tag post-processing improvements (see the discussion on the system's module components above, in particular, the explanation about the Inliner). The more capable the system is to handle tags, i.e. inline mark-up, the less disruptive tags are for the quality of the output and for the linguist in charge of the post-editing exercise. This was perceived and reported by the post-editors, who declared that having to perform fewer corrections or reorderings, the task could now be completed more quickly.

Apart from the time savings recorded, we have noticed significant resource reduction for a typical automotive project within the last two years. Whereas in 2008, 12-16 people (including 2-3 PMs / QA personnel) were required, only 4-7 people inclusive of PMs and QA personnel have been needed from the end of 2009 onwards.

More recently, between the April-July 2010 period, one of our senior PMs that is responsible for the coordination of ENES automotive projects has reported an average of 40% cost-savings in the payment for five large projects[19] thanks to the deployment of the PangeaMT AUT engine. She has also pinpointed that for the size range of these projects (approx. between 250,000 and half-a-million words) only 3 people are usually involved, herself as PM, and two post-editors (one of them being possibly more senior or experienced than the other).

One gold record as reported to her by one of these two linguists was that he reached a 25,000 word/day productivity, inclusive of fuzzy match revision. Happily enough, this finding is not isolated or exclusive of the AUT domain[20]. Consequently, in our training and communication exchange sessions with post-editors that represent different language directions and domains, we have established a target output per head of 50K words finalized in two consecutive days, particularly when post-editing output coming from a customized engine that has already gone through two or more retrainings.

One of our senior post-editors, mainly working with output from the Electronics and Computer Hardware – ECH engine, talks about the round number of + 1,000/hour in post-editing. In her first star project, she calculated an average of 1,375 words an hour during two working days. She still had time left to check the text and handle 30,000 words with a 100% match. The whole job comprised 51,000 words.

Needless to say that, thanks to PangeaMT's capabilities for outputting in open-standard formats, such as TMX or XLIFF, internal and external post-editors can perform their task in their environment of their choice. Unfortunately, what is preventing us from adopting AUT and other PangeaMT customized engines even more extensively at our end, thus not benefitting from time- and cost-savings as much as we could, is the fact that projects are sent out and requested in formats that are not industry standards. The complexity of the projects in terms of excessive folders or funny distribution of content (to translate) may also be perceived as a drawback, as maybe the time to extract, align or prepare the text for PangeaMT is longer than the estimated time saving connected with its use.

3.2 A few words about real-world customizations of PangeaMT

Typically, long-term corporate clients of ours, as LSP Company, who continue to have a large translation volume[21], year after year, have welcomed the idea of a customized PangeaMT to do more with less!

Sony Europe is a good example of that, a case study that has been widely presented in the last months in a number of industry events. The first ENES engine for Sony shares a similar motivation for engine creation and exploitation with the AUT engine described above, although its size at the time when these statistics were extracted is somewhat smaller. The next retraining is in fact scheduled in a few days only.

[19] Identified projects (P_534, P_11042, P_123, P_122, and P_426) amounted to 2,018,936 total words, out of which 125,071 were new.

[20] Post-editors of a forklift truck project machine translated by the ENES Technical - TEC PangeaMT engine have recently shown a high productivity level of 9,500 words in two days even if the topic was underrepresented in the training corpus. Thanks to our ENFR Software – SOF engine, a 27,000 word in 3 day productivity level has been reported, 3,5 days including QA.

[21] Dealing with 2-3 million words / year.

Training	Sentences	540,740
	Running words	5,897,779
	Vocabulary size	230,047
Test	Sentences	2,000
	Running words	21,692
	OOVs	466
	Perplexity (3-gram)	129.16

Figure 5. PangeaMT for Sony Europe engine training matrix (June 2010)

There were a number of challenges in relation to this customized development, such as the client's content lacking a uniform, standard format or being heavily formatted. It comes as no surprise that the localization manager there has been particularly appreciative of PangeaMT's I/O open-standard capabilities (TMX and XLIFF) and the Inliner. Pangeanic has provided Sony Europe with a solution web interface, which is password-protected, tracks user logs and gives the user the Tag-Optimal/Tag-Trans option discussed above.

An example of an interesting PangeaMT development for a new corporate client would be the ENDE solution for Sybase. The initial data volume for training was just 5 million words. No external data was added to guarantee total adherence with company's typical language register. However, if the engine is suddenly confronted with the style of a new product release literature, with longer sentences, the quality of the MT output will suffer.

Nevertheless, after only 2 months of using the solution's version one, productivity gains in the region of 50-300% were reported. This is quite an impressive figure, especially considering that the feedback being provided also guided us to improve some of our peripheral modules.

4 Conclusions & ongoing/further work

In this paper we have presented PangeaMT focusing mainly on its open-standard processing and generation capabilities.

Due to our own needs and those of our long-term corporate clients we needed to devise a translation automation strategy that encompassed relatively rapid development and domain-specific machine translation. Moreover, it was necessary that our solution range was capable of handling industry-wide, open standards to foster interoperability and avoid the undesired lock-in effect and other unjustified, painful costs.

We are also working in a technical procedure that will allow for engine retraining automation. This will be an incredibly powerful development feature that will allow clients and users to become independent from us even earlier, updating their systems at will and running BLEU scores to test improvement. They will just get back to us when in need of new engines for other domains/languages.

With regard to further scientific work areas, we are constantly testing and creating pre- and post-processing techniques with a view to making our PangeaMT engines more agile and accurate and tackling more language combinations and domains.

The recently improved pre-processing code now stores additional information for existing, advanced features and others that may be incorporated later. It also provides information, embedded in the pre-processed text, to help Moses identify which words make up a fixed phrase that should be translated as a single unit or be reordered in a certain way (reordering constraints: walls, zones, etc.).

In the case of customized developments that have proved particularly challenging beyond language (domain) specificity, but rather due to richness of uncommon symbols, newly created pre-processing techniques allow for embedded information to help the system restore symbols that are surrounding some expressions typical of client-specific content.

An exhaustive experimentation aimed at creating rules that make the pre-processing stage more generalist has enabled us to conceive a compact and sufficient command list that can tackle challenges from a variety of client- and domain-specific corpora; and to produce new and very complex regular expressions, which are much more powerful and can handle much more casuistry than before.

The post-processing code has also been thoroughly improved, some of the advances being:

- The matching calculation between eliminated and transformed text is much more efficient and robust;
- HTML/XML/TMX tags can now be incorporated more correctly than before, without having to fragment the sentence (which would lead to context loose and mistranslations).

- DNTs can be better categorised.
- The system's capabilities to eliminate and add white spaces, in an attempt to follow the spacing pattern of the sentence-to-translate, have also been improved.

Some of the scheduled improvements will be aimed at bettering what is working already. For instance, better capitalization handling in specific cases, better handling of DNTs, or profit from the alignment information at word level to restore inline tags and punctuation marks even better.

Acknowledgments

We are grateful to our long-term corporate and institutional clients who, having known us as a technology-oriented LSP, have entrusted us with the building of their customized PangeaMT engines. Pangeanic would also like to acknowledge the R&D work being accomplished in collaboration with our technology partner, the Instituto Tecnológico de Informática (ITI) at the Universidad Politécnica de Valencia.

References

Roni Rosenfeld. 2000. Two Decades Of Statistical Language Modeling: Where Do We Go From Here? *Proceedings of the IEEE*, 88: 1270-1278. http://www.cs.cmu.edu/~roni/papers/survey-slm-IEEE-PROC-0004.pdf

PangeaMT has been widely presented in major GILT and academic conferences throughout 2009/10. Please contact us if you should like to receive any of these materials.

Appendix A: Procedure to carry out Post-Editing (PE) of PangeaMT Output

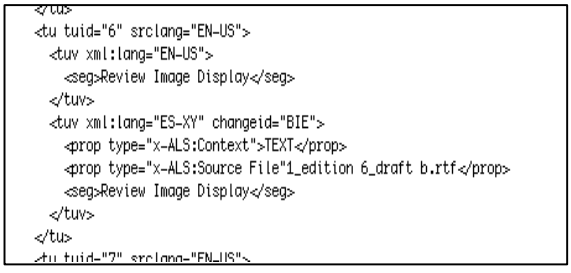

Figure 5. Input file in TMX format – fragment.

This is a description of the procedure we point our PangeaMT solution customers to when they wish to accomplish PE tasks on their own:

Step 1: Set your TM matching options to leverage everything above e.g. 75-90% match from the memory and send everything below that to your customized PangeaMT engine in TMX/XLIFF file.

Export your Project Translation Memory (preferably untranslated segments only). You won't pay for 100% matches, only for the real new text you need!

Your input file, for instance, in TMX format will look like figure 5. PangeaMT will then place a translated segment in the target segment like this one:

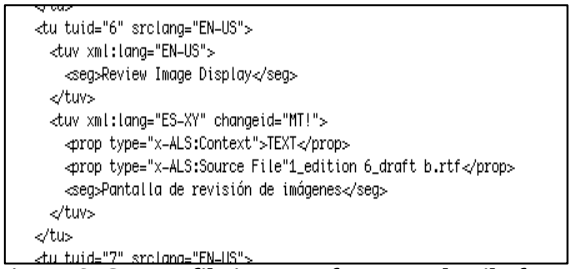

Figure 6. Output file in TMX format – detail of MT! indication tag.

Step 2: Now import the TMX into your translation system. Set a penalization for translator MT!.

Step 3: Start post-editing: your translation program will stop at untranslated segments over a 75-90% match, which requires little translation, and at those translated by MT! for quick post-editing.

With this kind of procedure, once more in line with our open-standard philosophy, we ensure that our clients keep working in their preferred[22] translation memory tool for data leverage.

It is important that the client understands that keeping track of how much data has been post-edited in a certain period of time is essential to arrange for system retraining. Post-edited material will lead to system maturity. Depending on language pair, domain specificity, etc., a system can reach maturity after 3-5 retrainings.

In other words, the system learns in time to translate better and increases its topic coverage and depth when retrained (updated, if you like) with more post-edited data. Inexpensive retrainings for fine-tuning will be usually scheduled at design stage although we remain open for suggestions like in the rest of the customized development stages.

[22] This also ensures a smooth transition for your freelancers as you provide them with quality, pre-translated output that they can easily post-edit in, for example, TagEditor or as .itd.