Processing Annotated TMX Parallel Corpora*

Rui Brito[1], José João Almeida[1], and Alberto Simões[2]

[1] Centro de Ciências e Tecnologias da Computação
Universidade do Minho, Braga, Portugal
{ruibrito,jj}@di.uminho.pt
[2] Centro de Estudos Humanísticos
Universidade do Minho, Braga, Portugal
ambs@ilch.uminho.pt

Abstract. In the later years the amount of freely available multilingual corpora has grown in an exponential way. Unfortunately the way these corpora are made available is very diverse, ranging from simple text files or specific XML schemas to supposedly standard formats like the XML Corpus Encoding Initiative, the Text Encoding Initiative, or even the Translation Memory Exchange formats. In this document we defend the usage of Translation Memory Exchange documents, but we enrich its structure in order to support the annotation of the documents with different information like lemmas, multi-words or entities. To support the adoption of the proposed formats, we present a set of tools to manipulate the different formats in an agile way.

Keywords: parallel corpora, annotated corpora, TMX

1 Introduction

Multilingual corpora [7] are very rich resources. They have been used for very different tasks like training machine translation software [5,4], extracting bilingual resources [18,8,19] or information retrieval [9,10].

Unfortunately there is no widely used standard to share parallel corpora at their raw level or with part-of-speech annotation. Some corpora are made available in specific XML formats together with simple programs to process them. Some others are made available in formats like the Text Encoding Initiative (TEI) or the XML Corpus Encoding Initiative (XCES). Unfortunately these standards are not flexible enough for the tasks they are being used, and therefore each user expand and/or interpret the standard by their will [17].

In this document we present a set of extensions to the Translation Memory Exchange (TMX) format to store annotated multilingual corpora. Our main guideline was that these formats should be easy to process using standard XML parsers, following the TMX schema, but not making it awkwardly difficult to

* This research has been carried out thanks to Portuguese National Funds, through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project PEst-OE/EEI/UI0752/2014.

parse. Instead of just describing the format, we will show a set of tools ready to process them. These tools are available as Open-Source Software and can be used and bettered by any user.

First, in section 2, we will briefly discuss the available formats for encoding parallel corpora. Then, in section 3, we will detail the *annotated translation memory exchange* (aTMX) and the *partially lemmatized translation memory exchange* (plTMX) formats, including some examples. Follows section 4 that presents the tools used to produce these formats, and section 5 that explains how to use our toolkit to process these formats. Finally, section 6 draws some conclusions and points different evolution directions.

2 Parallel Corpora Encoding Formats

There are a few standards to encode parallel corpora. The main problems [17] with these standards are the lack of documentation and evolution:

- The *Text Encoding Initiative* (TEI) is not devoted specifically for this purpose, and its way to encode parallel corpora is not versatile: parallel corpora are usually encoded in two different files, one for each language, and then a mapping file. This makes its processing error prone.
- The *XML Corpora Encoding Standard* (XCES) is outdated, unmaintained and incomplete. There are some researchers that still release their corpora in this format but, as the standard is silent regarding a lot of details, researchers tune the format to their will, making it hard to process.
- The *Translation Memory Exchange* format is quite simple to encode translation memories. As a sentence aligned parallel corpus can be seen as a translation memory this format has been used by some projects to encode parallel corpora. Nevertheless, it does not support, natively, any kind of mark-up to annotate the corpus.
- The *XML Localization Interchange File Format* (XLIFF) is specially used to store software localization translations. Just like TMX, it can be abused to store parallel corpora, but the XML overhead is bigger than using TMX.

Given the status of these formats there are some adaptations, just like ours, to known standards. For example, Forcada [3] proposes an idea similar to our, but extending the TMX tags at their limits. Although this gives extra flexibility to the annotation process, it makes it extremely difficult to keep track of the annotation. Also, the addition of XML tags for each word makes the document huge. Note that if a raw TMX for a parallel corpus can take up to 3 Gigabytes, adding annotations to each word using standard XML tags can make the file 3 to 5 times bigger.

Another Achilles' heel for the wide use of these formats is the lack of tools prepared to their manipulation.

3 The Annotated TMX Format

Annotated corpora can be powerful tools for developing and evaluating linguistic theories [6], forging a path for greater linguistic understanding and rigour. Annotations may include structural mark-up, part-of-speech (PoS) tagging, parsing, and numerous other representations.

3.1 Basic Format

As discussed previously the simpler formats that are being used are TMX and XLIFF. The first one is more known and therefore, there are more tools that deal correctly with it. This resulted in choosing TMX as the base format for our work. Figure 1 shows a two translation memory excerpt of a TMX file.

```
<tmx version="1.4">
<header creationtool="po2tmx" creationtoolversion="1.9.0"
            segtype="sentence" adminlang="en" srclang="en"/>
<body>
 <tu>
  <tuv xml:lang="EN">
   <seg>Display dialog boxes from shell scripts</seg>
  </tuv>
  <tuv xml:lang="PT">
   <seg>Apresentar caixas de diálogo a partir de scripts de consola</seg>
  </tuv>
 </tu>
 <tu>
  <tuv xml:lang="EN"> <seg>Type your password</seg>    </tuv>
  <tuv xml:lang="PT"> <seg>Introduza a sua senha</seg> </tuv>
 </tu>
</body>
</tmx>
```

Fig. 1. Example of a TMX file with two translation units.

The next decision is how to annotate the text inside each one of the TMX translation units. Our main goal when discussing this issue was to reduce the overhead of the annotation. With this in mind, and given that a lot of researchers use the Open Corpus Workbench [2] to encode their corpora we defined the *Annotated Translation Memory Exchange* format (aTMX) as a sort of fusion between the formats of both TMX and CWB. With this fusion we eliminate the need for an XML entry in each text line and another in each tag, making this format very economic. Figure 2 shows the annotated TMX for the translation units shown in Figure 1.

Note that, given the column-oriented approach, where each column represents a layer, it allows the user to add desired level of annotation. The most common

```
<tu>
  <tuv xml:lang="en"><seg><![CDATA[  <s>
    Display         display         NN
    dialog          dialog          NN
    boxes           box             NNS
    from            from            IN
    shell           shell           NN
    scripts         script          NNS
  </s>  ]]></seg></tuv>
  <tuv xml:lang="pt"><seg><![CDATA[  <s>
    Apresentar      apresentar      VMN0000
    caixas          caixa           NCCP000
    de              de              SPS00
    diálogo         diálogo         NCMS000
    a               a               SPS00
    partir          partir          VMN0000
    de              de              SPS00
    scripts         scripts         NCMP000
    de              de              SPS00
    consola         consola         NCFS000
  </s>  ]]></seg></tuv>
</tu>
```

Fig. 2. Translation unit from a aTMX file.

columns are *word*, *POS* and *lemma*, as they are the usual output of taggers. Syntactical anotation (treebank-like) can be easily used adding one or more columns for labeled dependency graphs or similar. The same approach is used in the CoNLL[3] data format, MaltParser [11], and others.

3.2 Region Annotation

One of the big problems with corpora annotation is the way XML forces tags to be properly nested. So, when annotations nest clearly, the proposed approach allows the use of user-defined tags. For example, Figure 3 shows how one can annotate multi-word expressions.

Note that the sentence tags ('*s*') and the multi-word expression tags ('*mwe*') are inside a CDATA section. This means they will be completely ignored by any TMX parser. But in the other hand, after retrieving the CDATA contents, they can be fed up to a XML parser for further processing. For other types of annotations, that can not be properly nested, different CQP layers (columns) can be used.

[3] CoNLL is the Conference on Computational Natural Language Learning, that often includes shared tasks, where data is made available in a specific format.

```
<tu>
  <tuv xml:lang="en"><seg><![CDATA[  <s>
    <mwe lema="text_view" pos="NP">
      Text      text       NN
      View      view       NN
    </mwe>
  </s>  ]]></seg></tuv>
  <tuv xml:lang="pt"><seg><![CDATA[  <s>
    <mwe lema="vista_de_texto" pos="NP00000">
      Vista     ver        VMP00SF
      de        de         SPS00
      Texto     texto      NCMS000
    </mwe>
  </s>  ]]></seg></tuv>
</tu>
```

Fig. 3. Extract from a aTMX file with multi-word annotation.

4 Input Tools

To produce an annotated TMX we need a tool to process the TMX file, and another one to produce annotations for each language segment. To process TMX files we use `XML::TMX` [1], a Perl module that is ready to deal with big TMX files whose Data Object Model (DOM) does not fit into memory. For the annotation we conduced several experiments with two different tools: *Apertium-Tagger* [15] and *FreeLing* [13,14].

The approach for each of these taggers is slightly different.

- The API for *FreeLing* is available to be used in Perl [16] which allows to use one or more languages at the same time. Therefore, the TMX is processed one translation unit at a time, where each language is fed to the language tagger (algorithm 1). This approach is useful for any tool that allows the use through an API.
- For the use of *Apertium-Tagger*, the TMX is processed previously, creating two different files, one for each language. These files are processed independently by the tagger, and then joined together in the resulting TMX file (algorithm 2). This approach is useful for external tools that do not export a simple API.

Note that meta-information is stored in the TMX header `prop` elements, like with columns and tags are present in the current file, for each specific language.

5 Output Tools

To make a specific format usable by third-parties it is very important to release software that can be used with the formats. In this section we present three tools that process annotated TMX files and produce different type of resources:

Algorithm 1: Tagging process using a library.

```
langs ← langs(TMX);
foreach segment ∈ TMX do
    foreach l ∈ langs do
        seg_l ← select_l(segment);
        tagged_l ← tag_l(seg_l);
    rebuildTU(tagged)
```

Algorithm 2: Tagging process using an external tool.

```
langs ← langs(TMX);
foreach segment ∈ TMX do
    id ← id + 1;
    foreach l ∈ langs do
        seg_l ← select_l(segment);
        save(id, seg_l, file_l);
foreach l ∈ langs do
    tag_l(file_l);
foreach id ∈ IDs do
    foreach l ∈ langs do
        segment_l ← fetch(id, file_l);
    saveTU(segments);
```

- codify the multilingual corpora into Open Corpus Workbench (OCWB);
- produce partially lemmatized translation memories (plTMX);
- extract probabilistic translation dictionaries (PTDs) taking into account words morphological information.

5.1 Exporting to CWB

The format used to annotate the corpora was taken from the OCWB format. This allows the direct importation of the annotated corpora into it. The Perl module `XML::TMX::CWB`[4] include a method to import a translation memory (being it annotated or not) into OCWB, and allowing the choice to import some specific languages only. This process includes the encoding of each language corpus and then the alignment import for every language pair. The module also supports the inverse operation, exporting the OCWB into an annotated TMX file.

5.2 Computing Lemmatized Dictionaries

One useful resource extracted from multilingual dictionaries are word alignments, like the ones extracted by Giza++ [12] or the Probabilistic Translation Dictionaries (PTD) extracted by NATools [18].

We are specially interested in the probabilistic translation dictionaries. These dictionaries compute relationships between words from the two languages that comprise a parallel corpus. Statistically, it is expected that this relationship maps words from a source language to their translations in a target language. A standard PTD entry is presented in Figure 4.

There are two big problems when computing PTD. The first one is related to certain linguistic constructs, like the use of auxiliary verbs, where the statistical nature of the algorithm will create relationships between the auxiliary verbs. The

[4] Available in `https://metacpan.org/pod/XML::TMX::CWB`

$$
\textit{imaginar} \begin{cases} \textit{image} & : 57.75\ \% \\ \textit{(none)} & : 3.99\ \% \\ \textit{imagining} & : 3.64\ \% \\ \textit{fathom} & : 3.63\ \% \\ \textit{wondered} & : 3.18\ \% \\ \textit{picture} & : 2.74\ \% \\ \textit{imagined} & : 2.54\ \% \\ \textit{conceive} & : 1.84\ \% \end{cases}
\qquad
\textit{imagine} \begin{cases} \textit{imaginar} & : 48.89\ \% \\ \textit{ideia} & : 4.15\ \% \\ \textit{imagina} & : 3.85\ \% \\ \textit{suponho} & : 3.85\ \% \\ \textit{imaginava} & : 3.79\ \% \\ \textit{imagine} & : 2.31\ \% \\ \textit{sabia} & : 1.55\ \% \\ \textit{imagino} & : 1.53\ \% \end{cases}
$$

Fig. 4. Two examples of entries from a standard PTD, generated from a TMX file, one mapping Portuguese words to English, and another from English to Portuguese.

second problem is related to unbalanced morphology complexity. For example, in Portuguese (and in most of the romance languages) a verb produces easily more than a hundred forms, but in English it will produce just half a dozen. This kind of relation will create lots of relations between a single form in English to a lot of Portuguese forms, with each of these relations having a very low probability.

To help in this alignment we can use annotated TMX. Given that this format includes annotations we can take advantage of them to reduce ambiguity and reinforce asymmetrical relations. This can be done at different levels:

- It is possible to use only lemmas. In this situation the huge amount of forms of verbs is not a problem, given they will be all replaced by the infinitive form. This will happen similarly for other word categories.
- Together with the lemmas we can add portions of its part-of-speech. For example, adding a prefix to specify the word category (noun, verb, adverb, adjective, etc), and therefore obtain translations for words when used in different syntactic contexts.
- Also, we can use the idea of partially lemmatized translation memories (that will be discussed in the next section) to obtain a mix of standard and lemmatized PTD.

As an application example consider the construction of a bilingual verbs dictionary bootstrapped by parallel corpora. Consider the following process:

1. Produce an annotated TMX file from a standard TMX file;
2. Collapse each word entry to a token that saves its part-of-speech;
3. Use NATools to extract a pair of probabilistic translation dictionaries;
4. Filter the resulting dictionaries to include only verbs.

The result of applying this process to a literary corpus is shown in Figure 5.

In fact, step two of this process can be useful for different tasks. We can collapse each word information in different ways, like above, adding a part-of-speech mark to the lemma. The next section introduces the concept of plTMX, a TMX file whose words are special tokens.

5.3 Partial Lemmatized Translation Memories

Sometimes it is useful to convert an annotated TMX to something more simple that can be processed easily as if translation units were traditional sentences,

$$v\_imaginar \begin{cases} v\_imagine : 44.47\ \% \\ v\_wonder : 10.81\ \% \\ v\_think : 4.41\ \% \\ v\_suppose : 0.76\ \% \\ v\_sense : 0.70\ \% \\ *v\_have : 0.58\ \% \end{cases} \qquad v\_imagine \begin{cases} v\_imaginar : 59.24\ \% \\ v\_supor : 2.29\ \% \\ v\_ver : 2.19\ \% \\ v\_pensar : 2.08\ \% \\ v\_descobrir : 0.14\ \% \\ *v\_ir : 0.04\ \% \end{cases}$$

Fig. 5. Probabilistic Translation Dictionary of verbs.

but keeping some morphological information. We tackled this problem defining the concept of *partially lematized* translation memories (plTMX). These translation memories follow exactly the TMX standard, but instead of including simple words, or even the CQP annotation syntax, it includes tokens that mangle together words or lemmas, and some details of part of speech. Figure 6 shows the translation unit from Figure 2 as a partially lemmatized translation unit.

```
<tuv xml:lang="en">
  <seg>v_display adj_dialog n_box from adj_shell n_script</seg>
</tuv>
<tuv xml:lang="pt">
  <seg>v_apresentar n_caixa de n_diálogo p_a_partir_de n_script de
       n_consola</seg>
</tuv>
```

Fig. 6. Example of a partially lemmatized translation unit.

In the example verbs, names and adjectives were replaced by the pattern $\langle pos + \_ + lemma \rangle$. Remaining words were kept unchanged. Of course that the way these substitutions are chosen depends highly on the specific purpose of the experiment.

Figure 5 presents a PTD extracted from a plTMX. Compare the result from the previous unprocessed PTD. In this situation we have a quite strong relation between the verb *imaginar* and *imagine*, instead of the several weak relations of all the verb forms. Also note that, removing verb forms gave space to other interesting word to appear.

These resources can be used to bootstrap monolingual verb dictionaries as well. Consider the composition of a PTD, that maps the Portuguese language into the English language, with the PTD that maps the English language into the Portuguese language. This process creates a pseudo-probabilistic set of synonyms. The information associated with *imagine, imaginar* is presented in Figure 7.

Starting form a *standard* aTMX, we can easily produce a plTMX (using the default available converters or adapting them to our necessities) and, based on that, build a set of reusable tools to produce scalable rich bilingual resources.

$$
imagine \begin{cases}
imagine & : 26.76\ \% \\
wonder & : 6.57\ \% \\
think & : 3.93\ \% \\
see & : 1.29\ \% \\
suppose & : 0.94\ \% \\
sense & : 0.31\ \% \\
*have & : 0.26\ \% \\
assume & : 0.24\ \% \\
watch & : 0.05\ \% \\
imply & : 0.05\ \% \\
*do & : 0.04\ \% \\
consider & : 0.04\ \% \\
look & : 0.02\ \% \\
find & : 0.02\ \% \\
discover & : 0.02\ \% \\
figure & : 0.01\ \%
\end{cases}
\qquad
imaginar \begin{cases}
imaginar & : 27.22\ \% \\
pensar & : 6.30\ \% \\
supor & : 1.13\ \% \\
ver & : 0.96\ \% \\
perguntar & : 0.93\ \% \\
saber & : 0.35\ \% \\
sentir & : 0.28\ \% \\
*ter & : 0.19\ \% \\
achar & : 0.14\ \% \\
perceber & : 0.08\ \% \\
descobrir & : 0.07\ \% \\
*haver & : 0.04\ \% \\
pressentir & : 0.03\ \% \\
calcular & : 0.03\ \% \\
notar & : 0.03\ \% \\
*ir & : 0.02\ \%
\end{cases}
$$

Fig. 7. Pseudo-Probabilistic Synonymous Set.

6 Conclusion and Future Work

In this paper we defended the necessity of a simple yet versatile format to store multilingual annotated corpora. In order to achieve this we suggested the use of the Translation Memory Exchange format blended with the Open Corpus Workbench column-oriented format. The result allows the annotation of corpora with few overhead of syntactic sugar.

We presented a pair of algorithms using two different approaches, FreeLing-lib based and using external taggers (like Apertium-Tagger), to produce this format.

When processing corpora in annotated TMX format we were able to take advantage of the linguistic information for different objectives, like the extraction of lemmatized probabilistic translation memories. In our experience the use of plTMX proved to be very important and effective because they allow the use of word-based tools over annotated TMX. plTMX recycles annotated sentences back to sentences.

References

1. Almeida, J.J., Simões, A.: XML::TMX — processamento de memórias de tradução de grandes dimensões. In: XATA 2007 — 5ª Conferência Nacional em XML, Aplicações e Tecnologias Associadas. pp. 83–93 (February 2007)
2. Evert, S., Hardie, A.: Twenty-first century Corpus WorkBench: Updating a query architecture for the new millennium. In: Proceedings of the Corpus Linguistics 2011 conference. University of Birmingham, UK (2011)
3. Forcada, M.: On the annotation of tmx translation memories for advanced leveraging in computer-aided translation. In: LREC'14. Reykjavik, Iceland (may 2014)

4. Forcada, M.L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J.A., Sánchez-Martínez, F., Ramírez-Sánchez, G., Tyers, F.M.: Apertium: A free/open-source platform for rule-based machine translation. Machine Translation 25(2), 127–144 (Jun 2011)
5. Koehn, P., et al.: Moses: open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions. pp. 177–180. ACL, Stroudsburg, PA, USA (2007)
6. de Marneffe, M.C., Potts, C.: Developing linguistic theories using annotated corpora. In: Ide, N., Pustejovsky, J. (eds.) The Handbook of Linguistic Annotation. Springer, Berlin (2014), to appear
7. Melamed, I.: Models of translational equivalence among words. Computational Linguistics 26(2), 221–49 (2000)
8. Morin, E., Prochasson, E.: Bilingual lexicon extraction from comparable corpora enhanced with parallel corpora. In: Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web. pp. 27–34. BUCC '11, ACL (2011)
9. Nie, J.Y., Simard, M., Isabelle, P., Durand, R.: Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the web. In: Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 74–81. SIGIR '99, ACM, New York, NY, USA (1999)
10. Nikoulina, V., Kovachev, B., Lagos, N., Monz, C.: Adaptation of statistical machine translation model for cross-lingual information retrieval in a service context. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics. pp. 109–119. EACL '12, ACL (2012)
11. Nivre, J., Hall, J., Nilsson, J.: Maltparser: a data-driven parser-generator for dependency parsing. In: Proceedings of LREC-2006 (2006)
12. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics 29(1), 19–51 (2003)
13. Padró, L., Stanilovsky, E.: Freeling 3.0: Towards wider multilinguality. In: LREC. pp. 2473–2479. European Language Resources Association (ELRA) (2012)
14. Padró, L.: Analizadores multilingües en FreeLing. Linguamática 3(2), 13–20 (December 2011)
15. Sheikh, Z.M.A.W., Sánchez-Martínez, F.: A trigram part-of-speech tagger for the Apertium free/open-source machine translation platform. In: Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation. pp. 67–74. Universidad de Alicante, Alicante (2009)
16. Simões, A., Carvalho, N.: Desenvolvimento de aplicações em Perl com FreeLing 3. Linguamática 4(2), 87–92 (Dezembro 2012)
17. Simões, A., Fernandes, S.: XML schemas for parallel corpora. In: XATA 2011 — 9ª Conferência Nacional em XML, Aplicações e Tecnologias Associadas. pp. 59–69. Vila do Conde, Portugal (1–2 June 2011)
18. Simões, A.M., Almeida, J.J.: NATools – a statistical word aligner workbench. Procesamiento del Lenguaje Natural 31, 217–224 (September 2003)
19. Tiedemann, J.: Recycling Translations – Extraction of Lexical Data from Parallel Corpora and their Application in Natural Language Processing. Ph.D. thesis, Uppsala University, Uppsala, Sweden (2003)