CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Universidade do Minho: RepositoriUM

XML Schemas for Parallel Corpora

Alberto Simões[1] and Sara Fernandes[2]

[1] Centro de Estudos Humanísticos, Universidade do Minho, ambs@ilch.uminho.pt
[2] Departamento de Informática, Univ. do Minho, sara.fernandes@di.uminho.pt

Abstract. Parallel corpora are resources used in Natural Language Processing and Computational Linguistics. They are defined as a set of texts, in different languages, that are translations of each other. Note that these translations do not need to cover the full document, as we might have sentences translated just on some of the languages.
When dealing with the process of sharing resources, recent years have bet on the use of XML formats. This is no different when talking about parallel corpora sharing. When visiting different projects in the web that release parallel corpora for download, we can find at least three different formats. In fact, this abundance of formats has led some projects to adopt all the three formats.
This article discusses these three main formats: XML Corpus Encoding Standard, Translation Memory Exchange format and the Text Encoding Initiative. We will compare their formal definition and their XML schema.

1 Introduction

Natural Language Processing and Computational Linguistics are examples of areas where corpora and, in particular, parallel corpora, are relevant resources. To best understand the concepts we will discuss, we should start by defining this concept.

The *corpus* (plural, *corpora*) term, born in Linguistics, refers to a finite collection of texts, usually from a restricted domain [5]. There are hundreds of examples of available corpora. The most well known is the British National Corpus[3].

A *Parallel Corpus* is a collection of texts in different languages, where each of them is a translation of each other. In some situations one of these languages is considered as the *source language*, and its translations as the *target languages*. While not consensual, it is usual to consider that a parallel corpus is aligned at the sentence level, meaning that there is a relationship between sentences (or, roughly, text sequences) in the different languages.

This alignment process is defined as: having two parallel texts, U and V, a sentence alignment of these texts is a segmentation of U and V in n segments, such that, for each i, $1 \leq i \leq n$, u_i and v_i are mutual translations, and u_i and v_i are, respectively, sequences of sentences from U and V [4].

[3] http://www.natcorp.ox.ac.uk/

June, 2011- Pages 59–69 ⋊ATA 2011 — ISBN: 978-989-96863-1-1

Note that this definition means that we might have segmentations u_i or v_i that are empty sequences from U and V. Therefore, there might exist sentences in one of the languages that does not have a corresponding translation. Indeed, the creation or removal of sentences during the translation process is common.

This definition can be expanded to a set of languages, instead of just a pair. In this situation, we have a set S of m texts T_i ($1 \leq i \leq m$), that have n segments each, such that, $\forall i, j \quad 1 \leq i \leq m \wedge 1 \leq j \leq n$, $t_{i,j}$ sequences of sentences are mutual translations.

The Parallel Corpora definition is nothing more than this mapping between segments in different languages. Researchers, being in the field of Natural Language Processing or Linguistics, like to enrich their parallel corpora with extra information. The kind of information to be added will highly depend on the corpus objective. Examples encompass the simple annotation of named entities (personal or company names, for instance), morphologic or part-of-speech tagging of each word, syntactic structure, etc.

This diversity of possible annotations makes it almost impossible to define a standard schema with all the alternatives one might want. Therefore, the adopted solution is the ability to define generic tags that each user can personalize.

In this article we will focus on three different formats that have been used by the research community to encode parallel corpora:

- The Text Encoding Initiative (TEI) schema (subsection 2.1);
- The Translation Memory Exchange (TMX) schema (subsection 2.2);
- The XML Corpus Encoding Standard (XCES) schema (subsection 2.3);

In the next section we will explain where they came from and the original purpose for which they were created. Their objectives are very different, which means that the level and type of annotation they can support is diverse. Nevertheless, they can all encode non-annotated parallel corpora, meaning it should be possible to define computational bridges to convert between these formats.

While section 2 will present each of these formats in particular, section 3 will compare their structure in means of usability and flexibility. Finally, section 4 discusses the directions users who need to encode parallel corpora should follow.

2 Parallel Corpora Encoding Standards

This section title is misleading, as just one of the formats (XCES, section 2.3) was developed specifically for XML corpora encoding.

All the formats we will discuss are currently being used by researchers to release parallel corpora and, some of these researchers, are making their corpora available in more than one format.

In this section we will not compare the schemas but, instead, define the subset that are relevant to encode parallel corpora and annotate possible language phenomena. Finally, we will perform a qualitative evaluation on their flexibility to encode parallel corpora (check section 3).

2.1 TEI: Text Encoding Initiative

The Text Encoding Initiative (TEI) collection of schemas [8] was created to help in preparation and interchange of electronic texts for most real-world situations. TEI is modular, and depending on the text being encoded the set of schemas to be used is different. TEI includes a big variety of schemas, to encode texts, verses, transcription of speech, standard dictionaries, lists of places and names (toponyms and onomastic indexes), tables, mathematical formulae, graphs, networks, trees and others.

In particular, TEI includes schemas to encode language corpora (chapter 15 of the TEI Guidelines for Electronic Text Encoding and Interchange) and for text segmentation and alignment (chapter 16).

All these schemas share a common schema, known as the TEI header. This header includes typical meta-information, as the name of the document, its authors, the document copyright, editor, publisher, year, etc. While meta-information is relevant when encoding corpora and parallel corpora, in this article we will be more interested in the means these schema have to encode the corpora, itself.

Nevertheless, we should stress the relevance of meta-information for corpora construction. It is very relevant to know the genre of the text (journalistic, literary, religious, etc), the age of the text (when it was written), its language and sub-languages, its type (oral, written), etc. All this information can be stored in the TEI header.

The macro-structure of a TEI corpus can be described as follows:

$$\begin{aligned} teiCorpus &\leftarrow teiHeader,\ (TEI \mid teiCorpus)^{+} \\ TEI &\leftarrow teiHeader,\ text \\ text &\leftarrow front\ ?,\ (body \mid group),\ back\ ? \\ group &\leftarrow (text \mid group)^{+} \end{aligned}$$

Note that this structure is quite rich. It is possible to have a header for the full corpora, and a separated header for each text. Also, each text might be grouped in different sections.

The *text* element is used by TEI to store all kind of texts. Therefore one can expect all kinds of mark-up to be possible inside this element. Although there are some corpus that might come from well structured data sources, most are processed by automatic tools, that just extract pure text. Therefore we can consider that a text is just a sequence of paragraphs (p element) or lines (l element, often used for verse lines).

Some texts include some other level of segmentation, like the *div* element, that is used to divide text into sections.

For text annotation, TEI provides elements below the line or paragraph level. It includes elements for sentences (s element), for clauses (cl element), phrases (phr element) and words (w element). In fact it provides elements below word level, as morpheme, character or punctuation character.

Given the amount of elements to annotate different levels of text, the annotation of a corpus in TEI format can be very detailed. Any one of these elements

can have attributes like *type* and *function* for phrases and clauses, *lemma* and *type* for words. Therefore, it is very simple to add all the needed information with these attributes, that have an open content type.

As for the alignment task, it is implemented as links between elements. Usually (but not necessarily), parallel corpora are encoded in TEI as three separate files: the text in the source language, the text in the target language, and the alignment file. This alignment file includes the usual TEI header, and a sequence of *linkGrp* elements. These elements have some meta-information, like the documents that are being linked (in the *xtargets* attribute), and includes a list of *link* elements. These elements can include a *type* attribute (that is usually the number of segments form the source-text and from the target-text that are being linked), and a *xtargets* or *targets* attribute that has the identifiers used in the individual text files for the *p* or *l* elements (although this mechanism makes it easy to link sub-paragraph parts, like sentences, clauses, phrases or even words).

As an example for a *linkGrp* element:

```
<linkGrp targType="head p" xtargets="jrc-pt;jrc-ro">
  <link type="1-1" xtargets="28;28"/>
  <link type="1-1" xtargets="30;30"/>
  <link type="1-1" xtargets="31;31"/>
  <link type="1-2" xtargets="32;32 33"/>
  <link type="1-2" xtargets="33;34 35"/>
</linkGrp>
```

More than two languages support is easy to perform, extending this mechanism. In fact, we can find two different solutions: first, instead of two text files, we have one per language, and instead of a *linkGrp*, we have a set of groups, one for each language pair; other solution is to have more than two fields in the *targets* or *xtargets* attributes.

This description on the TEI mechanisms for encoding corpora and their alignment wasn't very detailed as we do not intend to write a tutorial, but instead, to compare the formats. Therefore, we invite the interested reader to consult the Guidelines for Electronic Text Encoding and Interchange that are available on the web[4].

TEI is a very detailed schema. Therefore, there is mostly any kind of text that can not be encoded as a TEI XML file. The drawback is the leaning learning curve.

As an example of project/corpus encoded in TEI, please check the multilingual parallel corpus based on the *Acquis Communautaire*[5], the total body of European Union (EU) law applicable the the EU Member States [6].

[4] `http://www.tei-c.org/release/doc/tei-p5-doc/en/html/index.html`
[5] `http://wt.jrc.it/lt/Acquis/`

2.2 TMX: Translation Memory Exchange

The Translation Memory Exchange format was designed for the interchange of translation memories across different vendors of computer assisted translation (CAT) software. It is a standard, or norm, defined by the Localisation Industry Standard Association[6] (LISA). LISA is an association where some Universities and the major companies with CAT software or localization offices have a seat. Examples of partners are Abbyy, Adobe Systems, Autodesk, Cisco Systems, Dell, Hewlett-Packard, ICANN, Intel Corporation, Lucent Technologies, OASIS, SDL International, Skype, Trend Micro, VMWare and XEROX.

To understand the idea of translation memory it is helpful to explain how a CAT software works. When performing a translation task, the translator is faced with sentences already translated by herself or by someone on her group. Therefore, a CAT tool stores in a database all performed translations. These translations are stored sentence by sentence (or sequence of words by sequence of words, since the reuse of translations is more effective with short sequences of words).

Therefore, a translation memory can, in a simplified way, be seen as a set of pairs that relate sequences of words in two different languages. This informal definition is quite near the definition of parallel corpora. Note that for parallel corpora we are forcing an order, a thing that translation memories do not guarantee by themselves. Given that translation memories are stored in XML files an implicit order (the order of appearance) exists. This makes the TMX format relevant for storing parallel corpora.

There is a working draft on TMX version 2.0, dated of 2007. Unfortunately no developments have been done on this proposal. Therefore all CAT tools and researchers using TMX are using the 1.4b specification.

Again, please be aware that we are simplifying the structure of TMX removing elements not relevant for the purpose discussed here. The macro-structure of a TMX file is defined as follows[7]:

$$
\begin{aligned}
tmx &\leftarrow header,\ body \\
header &\leftarrow @creationtool,\ @segtype,\ @srclang,\ @adminlang, \\
&\qquad (note \mid prop)^{\star} \\
note &\leftarrow \#PCDATA \\
prop &\leftarrow @type, \#PCDATA \\
body &\leftarrow tu^{\star} \\
tu &\leftarrow @srclang,\ @segtype,\ ((note \mid prop)^{\star}, tuv^{+}) \\
tuv &\leftarrow @xml:lang,\ ((note \mid prop)^{\star}, seg) \\
seg &\leftarrow \#PCDATA
\end{aligned}
$$

[6] `http://www.lisa.org/`

[7] Attributes are denoted with the @ symbol. Also, the *seg* element definition is simplified.

Explaining, a TMX file is a header with some meta-information and a body with a sequence of translation units (*tu*). A translation unit is a sequence of translation unit variants (*tuv*) with a segment (*seg*). Figure 1 presents a simple TMX file.

```
<?xml version="1.0"?>
<tmx version="1.4">
  <header creationtool="XYZTool" creationtoolversion="1.01-023"
          datatype="PlainText"   segtype="sentence"
          adminlang="en-us"      srclang="EN" />
  <body>
    <tu>
      <tuv xml:lang="en"><seg>hello</seg></tuv>
      <tuv xml:lang="it"><seg>ciao</seg></tuv>
      <tuv xml:lang="pt"><seg>olá</seg></tuv>
    </tu>
    <tu>
      <tuv xml:lang="en"><seg>world</seg></tuv>
      <tuv xml:lang="en"><seg>earth</seg></tuv>
      <tuv xml:lang="it"><seg>mondo</seg></tuv>
      <tuv xml:lang="pt"><seg>mundo</seg></tuv>
    </tu>
  </body>
</tmx>
```

Fig. 1. Example of a simple TMX file.

Meta-information can be added at different levels. As the *prop* and *note* elements are open content they can be used mostly for everything. Also, as they can be added at different levels (*header*, *tuv* or *tu*) they make it easy to annotate specific translation units or units variants. Unfortunately there is not a way to aggregate translation units in blocks. This is a problem if you wish to tag each translation unit with the source where the text came from. With TMX we have only two options: create a different TMX for each text source or to tag each translation unit with the text source. If we had a way to create blocks we could associate that information to blocks.

Regarding word annotation, TMX files support is very poor or inexistent. It supports some in-line tags but only one can be barely used to annotate text. Its name is `hi`, standing for *highlight*, and has only two possible attributes: *type* and *x*. The first is for free use (and therefore the user can invent their own way to encode any desired information), and the second is used to match elements between translation units. That is, lets the user link words or segments between translations.

Of course one can add a namespace to the XML file to perform the annotation. In this article we are interested only on the native mechanisms each of these formats provide to the user.

The TMX file format is also being used to make available parallel corpora. As an example, check the OPUS[8] project [7], that includes different types of corpora to download in TMX format.

2.3 XCES: XML Corpora Encoding Standard

The XCES (XML Corpora Encoding Initiative) encoding specifications has been developed for and by the language engineering community, with the aim to provide guidelines for encoding various features in written text, morphosyntactic annotation, and alignment information, all of which are relatively stable and agreed-upon within the community.

XCES[9] is the instantiation of Corpus Encoding Standard (CES[10]) as an XML document. CES was developed when SGML (Standard Generalized Mark-up Language) was broadly used, which explains CES not being originally developed in XML. One of the main problems of XCES is being based on CES. Authors did not write documentation on XCES relying on CES documentation. Unfortunately, portions of the standard were changed and, based in the well known Murphy's Law, the way to encode alignments in XCES changed.

It follows the same concept of TEI. Instead of defining a single schema for encoding corpora, it defines a family of smaller schemas that can be combined together to achieve different kinds of annotation, accordingly with the user needs. This allows more flexibility on their use.

In this article we will look specifically to the schema designed to encode parallel corpora.

A formal view of the macro-structure of a XCES alignment document follows:

$$
\begin{aligned}
cesAlign &\leftarrow @fromDoc,\ @toDoc,\ @type,\ cesHeader,\ linkList \\
linkList &\leftarrow linkGrp^{+} \\
linkGrp &\leftarrow @fromDoc,\ @toDoc,\ @type,\ link^{+} \\
link &\leftarrow align+
\end{aligned}
$$

That is, an alignment document in XCES is divided in two main sections, just like most standards, an header with meta-data (that we will not dissect in this paper) and a body, named `linkList` where the relations between segments will be defined.

This `linkList` is usually divided in `linkGrp`, which are groups of alignments for a specific file. So, if our alignment document is specifying alignments among more than one pair of files, then the alignment document will have a `linkGrp` element for each document pair.

[8] `http://opus.lingfil.uu.se/index.php`
[9] `http://www.xces.org/`
[10] `http://www.cs.vassar.edu/CES/`

This schema should be possible to use in cases when only one pair of documents is being aligned (therefore with just one `linkGrp`) or cases when more than one pair are being aligned. The schema supports the attributes `fromDoc`, `toDoc` and `type` at two different levels, which makes it possible to define these attributes at the top level of the document, at the `cesAlign` element, emphasizing this information. The `fromDoc` and `toDoc` attributes are simple URI that point to the files being aligned. The `type` attribute specifies the type of alignment (paragraph, sentence, word).

Inside the `linkGrp` element, we will have each alignment information, in `link` elements. Unfortunately the documentation is missing, and the authors are not answering e-mail. This leads to a problem: the user needs to guess the semantic of the XML structure defined by the Schema.

The `link` element includes a sequence of `align` empty elements. The pointer from each `align` element to the text being aligned is performed using an `href` attribute. But no further information on how to fill in this element is given to the user. Also, given that `linkGrp` elements just have information to a pair of documents, it is quite strange that the `link` elements support more than two `align` elements.

Regarding the annotation of the documents, XCES has a detailed schema to annotate the documents structure. In fact, and although it is not as detailed as TEI, it includes a very good set of entities to encode paragraphs, lists, tables, images, poems, etc.

Finally, the word level annotation is obtained with yet another XCES schema. Unfortunately, this schema cannot be merged with the document annotation schema. Note that unlike TEI, where each schema can be imported in top of each another, as they all share the same root structure (you can see it as a super class, TEI base, and a set of instances, one for each type of document), XCES defines a complete new schema for each kind of information (document structure, alignment, and now word level annotation). They only share a header, where the meta-information can be added.

The main problem is that an word level annotation file (or, as XCES calls it, a chunk sequence file) is just a XCES header and a sequence of chunks. These chunks have linking information where the annotation can be *aligned* with the document itself (so, the document is stored in a file, the annotation in another file that includes information about what portions of the file are being annotated).

Each of these chunks, include a sequence of analysis (called *feat* in XCES documentation, probably as an abbreviation to feature and not the English word). These elements are a key/value pair, where the user can include the type of information he would like.

The main advantage of this approach is flexibility. The user can encode virtually anything, but it is not easy to maintain. Consider the annotation of part-of-speech for each word in a text (say, the type—verb, adverb, adjective, etc.—, genre, number and verbal tense). For each one of these properties a *feat* element will be needed. And for each word, a *chunk* element with the proper linking information will be required. This is totally inefficient for processing purposes.

3 Comparing TEI, TMX and XCES

As described in the previous section, these three formats are quite different, and they were designed for different objectives. Table 1 compares some of the most relevant features of these formats. Note that we are comparing them with parallel corpora encoding in mind. So, documentation refers to the documentation on how to use these formats to encode parallel corpora, and dedicated tools, the availability of tools to encode and manage parallel corpora using these formats.

Feature	**TEI**	**TMX**	**XCES**
Documentation	++	+	–
Schema simplicity to encode parallel corpora	–	++	–
Multi-language support	++	++	++
Sentence level alignment meta-data	–	++	–
Word level annotation	++	□	++
Dedicated tools	–	++	–
Availability of encoded corpora	+	++	–

Table 1. TEI, TMX and XCES comparison table (++ stands for pretty good, + for enough, and – for limited support. A □ is used when no support is present).

A final decision on what encoding schema to use will highly depend on your objectives. Some examples and decisions you might take:

- Your parallel corpora will be used as a translation memory for machine translation software. In this case, it is clear that TMX format should be chosen;
- You have a bunch of XML files that you would like to align at sentence level. in this situation, using TEI or XCES would be better suited, as you can just create independent alignment files that will retrieve the parts being aligned from the independent XML files.
- You are making available a multi-language corpora, in alignment pairs. Then, it is easier to release each language as a separate XML file, and independent alignment files for each language pair. This way, the user can clearly choose what file to download.

The decision will also be highly dependent on what tools are available to manage your files. As it is described in the table above, TMX is well served with tools to manipulate translation memories. From a wide range of computer assisted translation tools, to small GUI tools or even libraries, like `XML::TMX` [1]. TEI is quite served on tools when used as a schema to encode textual document. To manipulate parallel corpora there are just some few scripts developed by researchers that release their corpora in TEI format. Finally, XCES have been quite neglected in the last years. For example, the OPUS project, already mentioned, is trying to encode their texts in XCES. But they are following the CES documentation and using XML format. The lack of proper documentation is making this standard completely unusable.

4 Conclusions

In this article we gave a brief insight of the three major schemas available to encode parallel corpora. As the previous section showed, if we compare directly the features for each standard, we will end up selecting TEI as the best. It is not just well documented but it also includes in-depth discussion on the schema features. The biggest drawback is related to its embracing philosophy. As all kind of texts can be encoded in TEI it makes it quite difficult to develop robust tools that can handle the full schema.

The TMX format is in the other end of the continuum. It was developed for a specific purpose, it is very simple and fully functional for its main objectives. Being small, makes it quite easy to develop tools manipulating it[11]: all computer aided translation software have import/export facilities for this format.

XCES is in the middle. Is was designed for a specific purpose, but generic enough to embrace a bigger set of documents related with that purpose. Its main problems are related to the lack of documentation and lack of usage. In fact, some researchers claim they are releasing their corpora in XCES format, but they are just encoding CES in XML, and XCES is more than that.

How to chose one of them is a problem. But for sure, the authors do not recommend XCES. It lacks documentation, it is not implemented on any tool, no project adopted it and, more important, the authors are working on some other standard (GRaF [3]) and are not maintaining XCES anymore.

The biggest conclusion we can get from the analysis of these three standards is that the fact of a specific standard being developed and thought for a specific type of usage it does not mean that researchers will adopt it. There are two main details that are crucial for the community to adopt a specific schema:

- If it is somewhat complicated, it should be very well documented. If it is more simplistic, some lighter documentation should be enough. But, without any kind of documentation it is hard for any researcher to give credit to that schema.
- If the schema was defined by more than one person, and in special, was defined by teams of well known departments, it should mean that these teams are interested on it. Therefore, some results, comprising results and/or tools, should be available. These tools/results should be relevant enough to convince researchers to look to that specific schema.
- To define a proper XML schema is not enough to know the field that is being annotated. A proper formation on mark-up languages is indispensable.

Authors are convinced that these factors are the main factors for the current status of XCES.

In the Per-Fide project [2] one of the main goals was to make available all the constructed corpora in the three formats: XCES, TMX and TEI. After this analysis, the authors are targeting their tools only on TEI and TMX formats.

[11] In fact only 90% of the schema is really used on most tools, but this subset includes the most relevant features.

Acknowledgments

This work was funded by the project *Português em paralelo com seis línguas (Português, Español, Russian, Français, Italiano, Deutsch, English)*
grant `PTDC/CLE-LLI/108948/2008` from *Fundação para a Ciência e a Tecnologia*.

References

1. José João Almeida and Alberto Simões. XML::TMX — processamento de memórias de tradução de grandes dimensões. In José Carlos Ramalho, João Correia Lopes, and Luís Carríço, editors, *XATA 2007 — 5ª Conferência Nacional em XML, Aplicações e Tecnologias Aplicadas*, pages 83–93, February 2007.
2. Sílvia Araújo, José João Almeida, Alberto Simões, and Idalete Dias. Apresentação do projecto Per-Fide: Paralelizando o português com seis outras línguas. *Linguamática*, 2(2):71–74, Junho 2010.
3. Nancy Ide and Keith Suderman. GrAF: A graph-based format for linguistic annotations. In *Proceedings of the Linguistic Annotation Workshop*, pages 1–8, Prague, Czech Republic, June 2007. Association for Computational Linguistics.
4. I. Dan Melamed. *Empirical Methods for Exploiting Parallel Texts*. MIT Press, 2001.
5. Alberto Manuel Brandão Simões. Parallel corpora word alignment and applications. Master's thesis, Escola de Engenharia - Universidade do Minho, 2004.
6. Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and Dániel Varga. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *5th International Conference on Language Resources and Evaluation (LREC'2006)*, Genoa, Italy, 24–26 May 2006.
7. Jörg Tiedemann and Lars Nygaard. The opus corpus - parallel & free. In *Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 26–28 2004.
8. Edward Vanhoutte. An introduction to the TEI and the TEI Consortium. *Lit Linguist Computing*, 19(1):9–16, April 2004.