TEI-Conformant Structural Markup of a Trilingual Parallel Corpus in the ECI Multilingual Corpus 1

David McKelvie & Henry S. Thompson

Human Communication Research Centre, University of Edinburgh
2 Buccleuch Place, Edinburgh, Scotland. <eucorp@cogsci.ed.ac.uk>

Abstract

In this paper we provide an overview of the ACL European Corpus Initiative (ECI) Multilingual Corpus 1 (ECI/MC1). In particular, we look at one particular subcorpus in the ECI/MC1, the trilingual corpus of International Labour Organisation reports, and discuss the problems involved in TEI-compliant structural markup and preliminary alignment of this large corpus. We discuss gross structural alignment down to the level of text paragraphs. We see this as a necessary first step in corpus preparation before detailed (possibly automatic) alignment of texts is possible.

We try and generalise our experience with this corpus to illustrate the process of preliminary markup of large corpora which in their raw state can be in an arbitrary format (eg printers tapes, proprietary word-processor format); noisy (not fully parallel, with structure obscured by spelling mistakes); full of poorly documented formatting instructions; and whose structure is present but anything but explicit. We illustrate these points by reference to other parallel subcorpora of ECI/MC1. We attempt to define some guidelines for the development of corpus annotation toolkits which would aid this kind of structural preparation of large corpora.

1. Overview of the ECI Corpus

1.1. Brief History and Acknowledgements

The ECI arose as a result of a concern shared by a number of European researchers in computational linguistics that waiting for fully funded support for collection and distribution of non-English corpus material would mean waiting too long. This concern crystalised into action, modelled on the Association for Computational Linguistics (ACL) Data Collection Initiative, following a meeting in Pisa sponsored by the Network for European Reference Corpora (NERC) in 1992. The original call for contributions to the ECI described it as follows:

> The European Corpus Initiative was founded to oversee the acquisition and preparation of a large multi-lingual corpus to be made available in digital form for scientific research at cost and without royalties. We believe that widespread easy access to such material would be a great stimulus to scientific research and technology development as regards language and language technology. We support existing and projected national and international efforts to carefully design, collect and publish large-scale multi-lingual written and spoken corpora, but also believe it will be some time before the scientific and material resources necessary to bring these projects to fruition will be found. In the interim, a small and rapid effort to collect and distribute existing material can serve to show the way. No amount of abstract argument as to the value of corpus material is as powerful as the experience of actually having access to some in one's laboratory. We aim to make that experience possible very soon, at a very low cost.
>
> The ECI is carrying out the first phase of this activity on a purely voluntary basis, under the guidance of an ad-hoc steering committee.

The majority of the work of collecting materials and permissions and converting them into a consistent format has been done at the Human Communication Research Centre, University of Edinburgh and at ISSCO, University of Geneva, under the overall supervision of Henry S. Thompson and Susan Armstrong, respectively. In addition to the infrastructure support provided by these institutions, modest financial contributions were provided by the European Network for Language and Speech (ELSNET), the LDC (University of Pennsylvania) and NERC. The ACL and the Language Technology Group of HCRC provided loans.

1.2. How to Acquire the ECI/MC1 CD-ROM

The CD-ROM is available in the US from the Linguistic Data Consortium (LDC), for members of the LDC or those making a bulk purchase, and otherwise from ELSNET, 2 Buccleuch Place, Edinburgh EH8 9LW, SCOTLAND. The cost from ELSNET is £20 plus postage, handling and tax where applicable, on signature of the necessary User Licence Agreement. Information about ordering the corpus can be had from Leeann Jackson-Eve <leeann@cogsci.ed.ac.uk>. Further information about the contents or markup in the corpus can be obtained from the authors of this paper at <eucorp@cogsci.ed.ac.uk>.

1.3. Overview of the contents of the ECI/MC1 corpus

The ECI/MC1 corpus contains almost 100 million words in 27 (mainly European) languages. It is now available on CDROM for research purposes at a low price. It consists of 48 opportunistically collected sub-corpora marked up using TEI P2 conformant SGML (to varying levels of detail). The sub-corpora vary considerably in size, the larger corpora include:

GER03 German Newspaper texts from the Frankfurter Rundschau
July 1992 - March 1993
Provided by Universitaet Gesamthochschule Paderborn Germany
Approximately 34 million words.

FRE01 French Newspaper texts from Le Monde
September, October 1989, and January 1990
Provided by LIMSI CNRS, France
Approximately 4.1 million words

DUT02 Extracts from the Leiden Corpus of Dutch
(newspapers, transcribed speech, etc)
Provided by Instituut voor Nederlandse Lexicologie, Leiden, Holland
Approximately 5.5 million words

MUL05 International Labour Organisation reports of the
Committee on Freedom of Association 1984-1989.
Parallel texts in English, French and Spanish
Approximately 1.7 million words per language

The corpus contains a number of parallel multilingual corpora. This paper will concentrate on the markup of one of these, the trilingual MUL05 corpus of International Labour Organisation reports.

2. Markup of the ILO corpus

2.1. Conversion of data into standard ASCII text files

The ILO corpus originally came to us from the International Labour Organisation in the form of a backup tape containing word processor files. Reading the tapes and converting their format into standard UNIX format text files was a non-trivial operation, which was undertaken by Dominique Petitpierre (ISSCO) and David Graff (LDC). The details of this step are unimportant for the purposes of the present paper, except to note that this initial stage of processing is often necessary and is often made difficult by lack of easily available documentation and requires specially written software.

The corpus files originally used the Wang WP ASCII World Languages Character Set. This character set includes normal characters, underlined versions of these characters, plus some control characters to control rendition features such as centering lines, indentation, sub- and super-script characters. Following information obtained from Wang, ISSCO and LDC converted the files to use the ISO-LATIN-1 character set with SGML-style markup for the rendition features.

2.2. Conversion of control characters into markup

The texts contain some markup which describes the physical shape and position of the characters. They have been converted into SGML rendition attributes attached either to a structural division (<div>) or to a <hi> ... </hi> section where necessary. If more than one rendition is applicable to a section of text, then the rendition attribute of the section is formed by concatenating the different rendition values separated by "." eg

<hi rend=ul.centr> Means underlined and centered text

There are difficulties in converting a command based markup scheme into a structural markup scheme. In a command based scheme control characters/sequences are used to change the state of some formatting machine in a sequential fashion. Thus the sequence:

[bold on] aaa [italic on] [bold off] bbb [italic off]

is a legal sequence of instructions to the word processor. Such sequences were a common occurence in this corpus. Converting this to SGML markup in a straightforward way, leads to the badly nested structure:

<bold> aaa <italic> </bold> bbb </italic>

since SGML describes the structure of text in a hierarchical fashion, (We assume here that we prefer not to use SGML processing instructions to mark these font changes). This is a common problem when converting such WP texts into SGML. Corpus annotation toolkits should provide tools for converting such sequences into valid SGML markup.

Again, without a detailed understanding of the word-processor's operations it is not always straightforward to change its control codes into structural markup e.g. in the sequence (line breaks as in the original)

A line of text
[underline on]
[underline off]

the <ul> ... would appear to have no effect. In fact, it became clear that the underlining was active for the full defined width of the line, thus underlining the previous line of text. Thus the correct SGML markup was:

<hi rend=ul>A line of text

Another nice example of the difficulties of attempting SGML markup which captures the semantics of the texts is the following: Some section titles had every second character underlined, originally this would have the appearance of text with a broken line underneath it. By the time we had SGML marked this up, it looked like:

<hi rend=ul>A</hi> <hi rend=ul>s</hi>e<hi rend=ul>c</hi>
t<hi rend=ul>i</hi>o<hi rend=ul>n</hi> <hi rend=ul>t</hi>i<hi rend=ul>t</hi>
l<hi rend=ul>e</hi>

which is totally unreadable, fails to capture the semantics of the markup in a clear and descriptive way and makes searching for words in the text difficult. Instead we introduced and documented a new value of the rendition attribute and recoded this as:

<hi rend=ul2>
A section title
</hi>

Similar cases occur in newspaper texts where the first letter of the first word of an article is in bold font and the rest of the word in normal font. In this case we have placed the whole word in a <hi rend=first.letter.bold> element.

2.3. Text markup invariants

We take the approach that although TEI markup is the correct technique for text annotation, not all users of our corpus will use SGML to access this data. We thus tried to keep text and markup separate from each other. The bulk of the data provided observes what we call the Text/Markup Invariant:

> Every line in a data file is either all text or all markup, and a line is a markup line if and only if it begins with a left angle bracket (<). This makes restricting your processing to 'plain text' very easy – just look only at lines which begin with some character other than <.

2.4. Determining the logical structure of the corpus

The corpus came to HCRC from ISSCO in the form of 292 files named 'doc.<NNN>.iso'. This was the physical form of the corpus which appears to conform with the original division of the corpus into Wang word-processor files.

In contrast to the above, the logical structure of the corpus can be expressed as:

ILO	$\rightarrow$ CORPUS(eng) CORPUS(fre) CORPUS(spa)
CORPUS(LANG)	$\rightarrow$ VOLUME(LANG)+
VOLUME(LANG)	$\rightarrow$ ISSUE(LANG)+ SPECIAL-SUPPLEMENT(LANG)*
ISSUE(LANG)	$\rightarrow$ CONTENTS(LANG) REPORT(LANG)+

That is, the ILO corpus is made of three language-specific corpora, in English, French and Spanish. Each language-specific corpus is made up of a number of volumes each of which contains all of the publications in that language in a particular year. Each Volume consists of a number of issues (normally three) and a number of special supplements, each of which consists of a single publication. Each ISSUE contains a table of contents and a number of REPORTS. Special Supplements do not contain reports within them. Each ISSUE has a table of contents, covering all the reports in the issue.

We decided to take the single-language ISSUE as the basis for our re-organisation of the corpus, that is, all the material from a single issue would be placed together into a single computer file. The first stage of the reorganisation of the corpus was to collect all of the files which contained information from a single one-language issue into a single computer file. Fortunately, the information needed to do this was contained in header blocks at the front of each WP file. In general, the re-organisation of a corpus into meaningful pieces is not a straightforward operation, and requires an understanding of the contents of the data.

Since the division of the documents into original files cut across the hierarchical structure of the documents, in the new single issue files the locations of the original file breaks was marked with the SGML <milestone> tag as follows:

<milestone unit=file n="Original File Name">

Each of the 'doc.<NNN>.iso' files had a header block at the start. These header blocks were removed from the new versions of the files since the information they contained was made explicit in the new structure.

2.5. Markup of structural text divisions

The ILO documents were highly structured, the difficulty was in capturing this structure in SGML. The lowest clearly marked level of structure in the documents was the numbered paragraph. These are consecutively numbered from 1 in each report or appendix. They have been marked-up as

<divN n={original number} type=ILOpara>

The value of N reflects where the ILO paragraph is in the document structure and can vary, it is typically 6.

Above this basis of ILO paragraphs, higher level sections were constructed, partly by inspection of the texts and partly by comparison with the table of contents. These are marked up using

<divN type={SectionType}>

where N varies depending on the depth of the structure nesting and SectionType describes the semantics of the division e.g. "Report", "Case", "CaseRecommendation" etc. Fortunately, most section titles were marked syntactically, for example with bold fonts, and the reports follow a fairly constant structure. The biggest difficulty was in determining the nesting structure of the divisions and the types of the divisions.

We decided to use numbered divisions <div1> etc rather than unnumbered divisions e.g. <div type=...> because end tags can be added automatically, and navigating around the files is made easier. Also, in this corpus, the type of a division did not always correlate with its level of nesting.

Since the higher level divisions were not explicitly marked in the text, we made an effort to determine the type of a division from its title. This was only possible due to the stereotyped nature of the reports. For example, each case report had a section giving the recommendations of the committee. However these were not so easy to find. The final search pattern which we used to find such titles was:

> /The Committee's recommen?dations?/
> /The recommen?dations? of the Committee/
> /Recommandations? +du +comité/
> /Recomendaci(o|ó)n(es)? del Comit(é|e)/
>
> (? means the previous item is optional, + means the previous item can be repeated more than once, (a|b) means a or b)

As can be seen, it would have been difficult to determine these patterns without an exhaustive search of the corpus. In fact they were arrived at by a process of iterative refinement.

Occasionally divisions have been introduced below the ILO paragraph level. Normally however, below the ILO paragraph level we have only marked sections separated by a blank line with <p> ... </p>. These normally contain running text, but were also used for items in lists and headings in tables etc.

A fairly complex perl script was written to process the files. The new files were then checked and edited by hand.

Another complication was the presence of footnotes and references to footnotes. Footnotes and references to them were marked in a number of slightly different ways in the text. For example, some were marked by a special control sequence, others were marked simply by superscript numbers in running text. These different ways had to be found and their difference in meaning (if any) discovered. This was complicated by the existence of footnotes running over several pages and even footnotes inside other footnotes. In the case of superscript numbers, all but two such occurrences were in fact footnote references. The other two were of the form "m^2" and meant square metres! Again without careful checking of all occurrences, such things will go unnoticed. Finally we added SGML IDs to all notes and cross-referenced them to the references to the footnotes. By doing so, we uncovered some more notes which due to slight differences in their syntax had been missed.

2.6. Cleanup of texts

A fairly common error in these files was the occurrence of the letters "l" and "O" in contexts where it was clear that they should have been the numbers "1" and "0" respectively. This is also a common error in files which have been OCR scanned. It proved easy enough to write a simple pattern which matched such letters in numeric contexts and convert them to numbers. However, before this error was discovered, it caused some other patterns looking for e.g. paragraph numbers, to fail and hence cause further errors in the markup.

Line internal hyphenation (eg a word followed by a hyphen and a space) have been quietly edited either to a complete word or to a hyphenated word where appropriate. I.e. except where it is clearly part of the significant layout of say headers and/or tables or when it part of a multi-word compound e.g. 'two- or three-weekly'. Line final hyphenation has been left as is.

There is still a considerable amount of further markup which could be applied to this corpus. For example, we made no attempt to markup tables or lists in any special way. No dates or proper names, which are an important part of the information content of this corpus, were marked as such.

2.7. Alignment of structure across three languages

Once the individual files had been marked-up using TEI structural markup e.g. major textual divisions, titles and paragraphs, the different language versions were compared with each other.

Using simple perl scripts and UNIX tools the different language versions were compared and were edited to ensure that the structure of each version was the same. As far as possible, each language version of a document has the same SGML structure as far as <divN> elements are concerned.

This means that if there is a <divN> in (say) the English version of a report, then there will be a similar <divN> in the French and Spanish versions, containing equivalent text. It is not guaranteed that the language versions are parallel down to the SGML <p> level. In most cases a <p> in one language corresponds to a <p> in the other languages, but not always.

This process was complicated by the following facts:

- Equivalent sections were in different positions in different versions, in particular appendices and tables were sometimes placed inline and sometimes placed at the end of a file.
- Sometimes there were missing sections in one of the languages, or extra notes or draft sections existed in one language.
- Sometimes previous markup processing had missed a section or a section title in one version. For example, normally section titles are separated from their text by blank lines. In the Spanish texts, some paragraph titles were included in the paragraph text and not specially marked. This was not noticed until we compared them with the other language versions.

3. Conclusions

The markup of this corpus involved a great deal of inspection of the data, looking for common patterns, determining their meaning or lack of meaning and devising TEI conformant markup which captured this meaning. Then pattern-matching programs which could convert the existing data into the desired new form had to be written. These transformations often required an iterative process of making edits, checking the results and modifying the edits in order to capture the full range of variability in the texts.

For example, in Spanish, ordinal numbers are often followed by a superscript o , however sometimes this is represented as the character ^{o} and sometimes by the sequence <superscript>o</superscript>, i.e. different typists use different methods to achieve the same end.

Until all producers of large documents produce their files in a common SGML format, there will be a role for this kind of document markup. It is certainly of interest to create tools which speed up this process. However, it is not as simple as writing a pattern matching program and running it over the data.

In the first case, it is difficult to write patterns which match all and only the desired data. For large complex texts it is necessary to assume that any pattern will need refinement and correction until it correctly matches ones intuitive understanding and the nature of the text. One can either impose structure on the texts, e.g. *define* a number to be a sequence of digits and stick to this definition. Or one must be prepared to allow the actual data to redefine or refine ones definition of what a number is e.g. allow l for 1 mis-typings etc.

Secondly, SGML markup is inherently recursive - finite state languages (i.e. most pattern matching languages) - find this difficult or impossible to deal with in full generality.

Finally, although we started with a clear definition of the markup that we wanted to add and how we would do it, we found that after 600 megabytes of data that our ideas about markup had considerably changed in order to model the data. With every such change one has the problem of checking that previously marked up subcorpora are still conformant.

Our suggestions would be:

- There is a role for imposing textual invariants on the data, for example removing all line final white space or placing all markup on separate lines. Environments which make it easy to define such invariants and which help in checking/enforcing them are needed.
- There is likewise great scope for having a library of small software tools which handle common tasks e.g. hyphenation or reordering font changes to make valid SGML.
- However there will always be a need for a text processing environment where it is easy to write programs to check and alter texts. Each new corpus is different. It would be a great help to the corpus annotator if such an environment could look after the common features of text processing, such as backup of old versions and checking planned edits for 'dangerous' operations.
- The ECI/MC1 corpus was annotated using mainly perl, emacs and various UNIX tools. This provided a reasonable text processing environment, but it could be improved. Describing how it could be improved soon descends to a long list of disconnected facts, for example, that "wc" (UNIX tool for counting words) does not work correctly on files containing 8-bit characters or that (unless one tweaks an obscure variable) Emacs will change the case of letters in edits in some contexts. It tries to be helpful, but sometimes gets it horribly wrong. Designing a text processing toolkit which is powerful, intelligent and transparently clear in its operation is needed, but almost certainly a pipe dream.
- Given this, we need better tools for searching and scanning data to find anomalous patterns in the data.

References

[1] "Practical SGML", Eric van Herwijnen, Kluwer Academic Publishers, 1990.

[2] "The SGML Handbook", Charles F. Goldfarb, Clarendon Press, 1990.

[3] "Programming perl", Larry Wall and Randal L. Schwartz, O'Reilly and Associates, Inc. , 1991.

[4] "Emacs users guide", Online documentation available with emacs.

[5] "Tutorial on Text Corpora", Mark Liberman and Mitch Marcus, ACL '92.

[6] "Guidelines for Electronic Text Encoding and Interchange", C.M.Sperberg-McQueen and Lou Burnard (eds), Pre-publication P2 drafts, Text Encoding Initiative, 1993. [Available from the List-serv list TEI-L in electronic form]