

Massive multi lingual corpus compilation: Acquis Communautaire and totale

Tomaz Erjavec,^{*} Camelia Ignat,[†] Bruno Pouliquen,[†] Ralf Steinberger[†]

^{*}Department of Knowledge Technologies
Jožef Stefan Institute
Jamova 39, SI-1000 Ljubljana, Slovenia
<http://kt.ijs.si/>, tomaz.erjavec@ijs.si

[†]European Commission - Joint Research Centre
I – 21020 Ispra (VA), Italy
<http://www.jrc.it/langtech/>, Firstname.Lastname@jrc.it

Abstract

The paper discusses the compilation of massively multilingual corpora, the EU ACQUIS corpus, and the corpus annotation tool “*totale*”. The ACQUIS text collection has recently become available on the Web, and contains EU law texts (the Acquis Communautaire) in all the languages of the current EU, and more, i.e. parallel texts in over twenty different languages. Such document collections can serve as the basis for multilingual parallel corpora of unprecedented size and variety of language, useful as training and testing dataset for a host of different HLT applications. The paper describes the steps that were undertaken to turn the text collection into a linguistically annotated text corpus. In particular, we discuss the harvesting and wrapper induction of the corpus, and the usage of its annotation with EuroVoc descriptors. Next, the text annotation tool “*totale*” which does multilingual text tokenization, tagging and lemmatisation is presented. The tool implements a simple pipelined architecture which is, for the most part, fully trainable, requiring a word-level syntactically annotated text corpus and, optionally, a morphological lexicon. To train *totale* for seven different languages we have used the MULTEXT-East corpus and lexicons; we describe this resource and the training of *totale*, and its application to the ACQUIS corpus. Finally, we turn to the current experiments in aligning the corpus, and developments we plan to undertake in the future.

1. Introduction

Parallel corpora (McEnery et al. 1997; Armstrong et al. 1998; Dimitrova et al. 1998; Koehn, 2002; Erjavec, 2004) are a prime resource for the development of multilingual language technologies. Serving as training datasets for inductive programs, they can be used to learn models for machine translation, cross-lingual information retrieval, multilingual lexicon extraction, sense disambiguation, etc. The value of a parallel corpus grows with the following characteristics:

- *Size*: larger corpora give not only statistically more reliable counts, but also reveal phenomena that are completely lacking in smaller samples
- *Number of languages*: the utility here grows quadratically with the number of languages, as each language can be paired with any other. While bi-lingual corpora usually contain at least one ‘major’ language, larger multilingual collections will also contain pairings of less common languages, where such a resource is of great value (Maltese-Finish for example).
- *Linguistic annotation*: can be used as a normalisation step on the raw text, hence reducing the complexity (search space) of the LT task; or for enabling multiple knowledge of the text (e.g. morphosyntactic tags, collocations, predicate-argument structure) to be exploited.
- *Semantic annotation*: refers to the classification of documents (or their parts, e.g. words) into some hierarchy of concepts, which can be used to access the data (e.g. the Semantic Web paradigm)

This paper discusses the compilation of a large, massively multilingual corpus that is tokenised, tagged for word-level syntactic information, and where each document is classified according to a rich ontology.

Additionally, we discuss the main tool that is being used and developed for the linguistic annotation of the corpus

The rest of this paper is structured as follows:

- Section 2 introduces the EU ACQUIS text collection and its harvesting and resulting corpus format
- Section 3 describes the text annotation tool “*totale*”, a trainable program, which performs multilingual text tokenization, tagging and lemmatisation.
- Section 4 introduces the MULTEXT-East dataset, which was used to train *totale* for seven languages.
- Section 5 turns to the current experiments in aligning of the ACQUIS corpus, and
- Section 6 gives the conclusions and discusses future work.

2. The EU ACQUIS parallel corpus

The core EU law, variously known as the Acquis Communautaire, is comprised of 8 to 82 million running words of texts depending on the language. This collection of documents, some dating back to the 1950s, has been for a while translated into the eleven languages of the ‘pre-enlargement’ EU. For the last six years, the candidate countries have been translating them into their languages – this was one of the conditions to enable their accession to the EU. This process has by now been mostly completed, and, what is more, the complete set of documents has been recently made available in HTML on the Web, at <http://europa.eu.int/eur-lex/lex/en/index.htm>, or, in Word format, for the translations to new languages at <http://ccvista.taiex.be/>.

Such a text collection is unprecedented in terms of size, the number of languages involved and availability,

Language	Number of texts	Length		Word count	
		Body text total size millions chars	Average size of texts	Total number of words (millions of)	Average number of words per text
English	40565	460	11347	76	1864
Spanish	40336	498	12355	82	2022
Finish	40000	319	7986	37	928
French ^(*)	40000	393	9821	66	1645
Hungarian	7613	69	9000	10	1270
Italian ^(*)	40000	356	8897	55	1381
Lithuanian	7534	62	8253	9	1164
Latvian	7939	62	7793	9	1113
Maltese	6140	55	8885	9	1469
Dutch ^(*)	40000	359	8981	55	1374
Polish	7782	68	8705	10	1242
Portuguese ^(*)	40000	361	9018	60	1491
Slovak	6968	58	8285	9	1276
Slovene	7895	59	7468	9	1179
Swedish ^(*)	40000	338	8439	51	1274
Czech	7120	56	7837	9	1222
Greek ^(*)	40000	422	10546	65	1636
German ^(*)	40000	378	9460	53	1322
Estonian	7865	61	7787	8	1029
Danish	40419	444	10995	65	1620

Table 1 Size of the corpus

^(*) expected size (not all were downloaded at the time we write these lines)

being freely available on the Web.¹ Furthermore, each of the texts has also been manually classified according to the EuroVoc thesaurus, at <http://europa.eu.int/celex/eurovoc/>, a large multilingual “ontology” being used by the Commission of the EU. This text collection could serve as the basis of an extremely useful large massively multilingual corpus, where each document is assigned to categories from a widely used ontology. Such a corpus could thus be useful not only for various machine translation researches, but also for “Semantic Web” experiments in, say, automatic descriptor assignment, i.e. document classification (Pouliquen et al., 2003), or cross-lingual document similarity (Pouliquen et al., 2004).

It is for these reasons that we proceeded with compiling the ACQUIS corpus, the process which consisted of the following steps:

1. *downloading* the texts: the interface enables locating the texts via their CELEX ID (unique identifier given for every EU official document); the copying was then a matter of querying over these IDs for all the languages; however, not all documents (IDs) are translated into each language, so the size of the various language parts varies considerably
2. *language identification* on the documents: for a few percent of documents, text purportedly in one language is in fact untranslated English text – such cases are not made part of the corpus;
3. *wrapper induction*: the texts can be usefully decomposed into the title, body of the text, the signature (e.g. “Done at Brussels, 24 September 1989, for the commission”, etc.), and annexes (containing tables or lists of codes, usually not

□

¹ For a similar corpus to ours see EUROPARL (Koehn, 2003), which, however, contains less languages although more text per language, and is not indexed with EuroVoc descriptors.

translated in all languages). It is the body that will contain most of the ‘useful’ text, yet the back-matter can comprise a considerable portion of the documents. These divisions were identified by Perl regular expressions over the texts, and the resulting “level 0” corpus was stored as XML

4. *linguistic annotation* of the texts: sentence, word and punctuation tags were added to the corpus, and the words given their context disambiguated lemma and morphosyntactic attributes; this processing, so far only for a limited number of the language components of the corpus, was performed by the program *totale*, described in the next Section
 5. *paragraph alignment*: paragraphs were given IDs, and (initial) alignment files made over language pairs of documents; current experiments are described in Section 5.
- The compiled corpus contains 20 languages, each text being an average of 7500 to 12300 characters, and 900 to 2000 words (depending on the language, see table 1).

3. Multilingual tokenisation, tagging, and lemmatisation

Corpora can be annotated with various linguistic annotation, such as syntactic structure, anaphora and their referents, terms, names, etc., but the basic steps for all such annotations are usually taken to be the following:

1. tokenisation
2. part-of-speech tagging
3. lemmatisation (or stemming)

We have developed a tool, named *totale* that performs the above steps in a multilingual setting. The program, written in Perl, implements a simple pipe-lined architecture, where plain Unicode (UTF-8) text is first tokenised, the word tokens (word-forms) then tagged with their context-disambiguated part-of-speech, or, more, accurately, morphosyntactic description (MSD), and the word-forms, given their MSD, lemmatised to arrive at the canonical form of the word.² The program can produce the output in several formats, in particular a in tabular form or encoded in TEI-compliant XML.

In Figure 1 we give a sample invocation of the program. The tabular output consists of four columns: the first lists the tokens as they appear in the input text; the second contains the token type or the tag marking the end of the sentence or other recognised structure; the third the lemmas of the words; and the fourth their MSDs. The second example invocation shows that the program can also produce XML formatted output.

The program is – once started – reasonably fast, i.e. it processes cca 100k words per minute. Starting time, however, is a problem. Partially this is to do with the system architecture of file-mediated sequential processing, and is partially due to the lemmatisation module for a language (with its possibly thousands of rules and exceptions) being loaded statically at the start of the program. The program is available for on-line experimentation at on <http://nl2.ijs.si/analyze/>.

□

² Note that lemmatisation, although similar to stemming, is a different and in general more complex task, as the result must be another surface word-form, e.g. the infinitive for verbs. While there is seldom much difference for English, other, inflectionally richer languages exhibit great variation in form between the word in the text and its lemma.

```

$ totale -l en
Doctor, can you help?
^D
      <TEXT>
Doctor  TOK      doctor  Ncfs
,       PUN
can     TOK      can      Voip
you     TOK      you     Pp2
help    TOK      help    Vmn
?       PUN_TERM
      <S/>
      </TEXT>

$ totale -l sl -f xml
Kapucini in zdravniki s kljunaškimi maskami na
obrazih so se znenada pojavili na vseh koncih
in krajih.
^D
<text>
<w lemma="kapucin" ana="Ncmpn">Kapucini</w>
<w ana="Ccs">in</w>
<w lemma="zdravnik" ana="Ncmpn">zdravniki</w>
<w ana="Spsi">s</w>
<w lemma="kljunaški"
  ana="Aopfpi">kljunaškimi</w>
<w lemma="maska" ana="Ncfpi">maskami</w>
<w ana="Spsl">na</w>
<w lemma="obraz" ana="Ncmpl">obrazih</w>
<w lemma="biti" ana="Vcip3p--n">so</w>
<w ana="Px-----y">se</w>
<w ana="Rgp">znenada</w>
<w lemma="pojaviti" ana="Vmpps-pma">pojavili</w>
<w ana="Spsl">na</w>
<w lemma="ves" ana="Pg-mpl----a">vseh</w>
<w lemma="konec" ana="Ncmpl">koncih</w>
<w ana="Ccs">in</w>
<w lemma="kraj" ana="Ncmpl">krajih</w>
<c type="TERM".</c>
<s/>
</text>

```

Figure 1: Output of totale

In the rest of this section we explain the three annotation modules of totale.

The tokenisation module

The multilingual tokenisation module `mlToken` is written in Perl, and, in addition to splitting the text input string into tokens has also the following features:

- Assigns to each token its token type. The types distinguish not only between words and punctuation marks but also mark digits, abbreviations, left and right splits (i.e. clitics, e.g. 's', enumeration tokens (e.g. a)) as well as URLs and email addresses
- Marks end of paragraphs, and end of sentence punctuation, where sentence internal periods are distinguished from sentence final ones.
- Preserves (subject to a flag) the inter-word spacing of the original document, so that the input can be reconstituted from the output – this consideration is important when several tokenisers are applied to a text, either for evaluation or production purposes.

The model for our tokeniser was `mtseg`, the tokeniser (and segmenter) developed in the MULTEXT project (Di Cristo, 1996); as with `mtseg`, `mlToken` also stores the language dependent features in resource files, in the case of `mlToken` of abbreviations and split/merge patterns. In the absence of a certain language resource, the tokeniser uses default resource files – in order to achieve best results, however, resource files for a

language have to be written – this task is helped by having pre-tokenised corpora for the language.

The tagging module

For tagging words in the text with their context disambiguated morphosyntactic annotations we used a third-party tagger, namely TnT (Brants, 2000), a fast and robust tri-gram tagger. TnT is freely available (but distributed only in compiled code for Linux), has an unknown-word guessing module, and is able to accommodate the large morphosyntactic tagsets that we find in various EU languages.

The tagger uses two resources, namely a lexicon giving the weighed ambiguity class for each word and a table of tri-grams of tags with weights assigned to the uni-, bi-, and tri-grams; examples of 4 words (*'dream-like'*, 2x *'breath'*, and reflexive pronoun *'se'*) and 4 tri-grams are given in Figure 2.

```

sanjsko :6
  Aopfsa:2 Aopfsi:1 Aopnsa:1 Aopnsn:1 Rgp:1
sape:4
  Ncfpa:1 Ncfpn:1 Ncfsg:2
sapo:10
  Ncfsa:9 Ncfsi:1
se:2031
  Px-----y:1967 Px---a-ypn:64

Px-----y:2226
  Vcps-sma:4
    Vmps-sma:2
    Rgp:2
  Vcip3s-n:794
    Vcps-sma:2
    Vcip3s-n:1
    ,:72
    Aopmsn:2

```

Figure 2: Tagger lexicon and MSD n-grams

Both resources are acquired from a correctly annotated corpus, where the induced lexicon can of course also be further upgraded.

The lemmatisation module

Automatic lemmatisation is a core application for many language processing tasks. In inflectionally rich languages, such as Slovene, assigning the correct lemma (base form) to each word in a running text is not trivial, as, for instance, nouns inflect for number and case, with a complex configuration of endings and stem modifications. The problem is especially difficult for unknown words, as word-forms cannot be matched against a morphological lexicon.

For our lemmatiser we used CLOG (Manandhar et al., 1998, Erjavec and Džeroski, 2004), which implements a machine learning approach to the automatic lemmatisation of (unknown) words. CLOG learns on the basis of input examples (pairs word-form/lemma, where each MDS is learnt separately) a first-order decision list, essentially a sequence of if-then-else clauses, where the defined operation is string concatenation. The learnt structures are Prolog programs, but in order to minimise interface issues we made a converter from the Prolog program into one in Perl. In the final instance the usage for determining the

lemma is simply the result of the function call `$lemma = lemmatise($msd,$wordform)`;

This function then calls the appropriate rule-set, which transforms the input wordform into its lemma. We give in Figure 3 an example of an induced rule for the Slovene MSD denoting the feature structure

```

$sub{'Afcfda'}='SUB_afcfda';
sub SUB_afcfda {
  my $w = $_[0]; my $lem;
  if ($w=~/(.*)svetlej#353i$/) {$lem=$1."svetel"}
  elsif ($w=~/(.*)polnej#353i$/) {$lem=$1."poln"}
  elsif ($w=~/(.*)b#353i$/) {$lem=$1."b"}
  elsif ($w=~/(.*)ejej#353i$/) {$lem=$1."el"}
  elsif ($w=~/(.*)ivej#353i$/) {$lem=$1."iv"}
  elsif ($w=~/(.*)anej#353i$/) {$lem=$1."an"}
  elsif ($w=~/(.*)kej#353i$/) {$lem=$1."ek"}
  elsif ($w=~/(.*)tej#353i$/) {$lem=$1."t"}
  elsif ($w=~/(.*)i#382ji$/) {$lem=$1."izek"}
  elsif ($w=~/(.*)enej#353i$/) {$lem=$1."en"}
  elsif ($w=~/(.*)rej#353i$/) {$lem=$1."er"}
  elsif ($w=~/(.*)nej#353i$/) {$lem=$1."en"}
  else {$lem="???"}
  return $lem;
}

```

Figure 3: An induced lemmatisation rule in Perl for the Slovene MSD:

PoS:Adjective, Type:qualificative
 Degree:comparative, Gender:feminine,
 Number:dual, Case:accusative.

4. MULTEXT-East resources

The main feature of *totale* is that it is multilingual and trainable for new languages, as the models for tagging and lemmatisation are induced from data. However, in order to make the tool useful, we first have to obtain such data, namely morphosyntactically annotated corpora and lexicons. It is an added advantage if the multilingual training resources all follow the same guidelines for tagset and corpus annotation design.

The MULTEXT-East language resources, a multilingual dataset for language engineering research and development, first developed in the scope of the EU MULTEXT-East project, have now already reached the 3rd edition (Erjavec, 2004). MULTEXT-East is a freely available standardised (XML/TEI P4, (Sperberg-McQueen, and Burnard, 2002)) and linked set of resources, and covers a large number of mainly Central and Eastern European languages. It includes the EAGLES-based morphosyntactic specifications, defining the features that describe word-level syntactic annotations; medium scale morphosyntactic lexicons; and annotated parallel, comparable, and speech corpora. The most important component is the linguistically annotated corpus consisting of Orwell's novel "1984" in the English original and translations.

For training *totale* we used resources for the Czech, English, Estonian, Hungarian, Romanian, Serbian, and Slovene. The MULTEXT-East *mtseg* resource files were used as sources for the mlToken resource files; the annotated corpus for training the TnT tagger; and the lexicons to improve the performance of the tagger and for training the CLOG lemmatiser.

While training the tagger on this data is very fast, training the lemmatiser is much more process intensive, as each MSD is learned separately – so, for Slovene or Czech, this meant learning more than 1000 different classes for a language, and the training time is measured in days.

The final paper will give the precise measures for various counts of the training sets, in particular: training corpus in tokens, types, lexical entries, wordforms, lemmas, MSDs.

5. Alignment

We have so far performed an experiment in language independent paragraph alignment of the English-Slovene pair, using the Vanilla aligner (Danielsson and Ridings, 1997). This aligner implements dynamic time warping by comparing the character counts of possibly aligned sentences (Gale and Church, 1993). The aligner is given the two files split into hard regions, which have to match among the files (in our case each document text corresponds to one hard region, and soft regions which are aligned 0-1, 1-0, 1-1, 2-1, 1-2, and 2-2. Soft regions are typically sentences, but in our case paragraphs, which, do, however, tend to be rather short corresponding to one or two sentences or even partial sentences.

An evaluation of the results showed that:

- The alignment is complicated by the fact that some English documents on the Web are previous versions of the ones that served as the source for the translation. The size of the amendments in terms of text percentage is usually not that large, but it does raise the error rate of the aligner significantly.
- The number of 1-1 links among the paragraphs is approx 90%. As these links are highly reliable, this means that, with an added heuristic or two, it would be simple to achieve (almost) 100% precise alignments at the cost of sacrificing approximately one fifth of the text, i.e. settling for 80% recall. This still leaves ample text for the aligned corpus.
- It would be relatively easy to introduce a pre-processing step that would take into account enumeration tokens (e.g. *I*, *a*, ...) and declare them as the hard regions for the aligner. This would most likely significantly localise and thus reduce the alignment errors.

6. Conclusions and further work

The paper has presented the compilation of massively multilingual corpora, in particular the EU ACQUIS corpus, and the *totale* tool used to annotate it. In addition to being massively multilingual and of significant size, the text collection also has the advantage of being rather freely available – the EU pages state that copying is allowed (if attribution is given). However, re-distribution is not allowed, so we would be ready to share our corpus with partners for research purpose, as far as they get the permission from the EU publication office (see http://europa.eu.int/eur-lex/lex/en/editorial/legal_notice.htm).

The other contribution is the text annotation tool *totale*, which has been trained for seven languages; however, we plan to expand the range of available languages, by using annotated corpora and lexicons for new languages (say from ELRA or LDC) to train the tagging and lemmatisation modules of *totale*.

Acknowledgements

The work presented in this paper was in part supported by the EU projects PASCAL and ALVIS, and by the first author's stay at the JRC.

Interchange, the XML Version of the TEI Guidelines.
The TEI Consortium, <http://www.tei-c.org/>

References

- Armstrong, S., Kempen, M., McKelvie, D., Petitpierre, D., Rapp, R., and Thompson, H. (1998). Multilingual Corpora for Cooperation. First International Conference on Language Resources and Evaluation, LREC'98. pp. 579-980. ELRA, Paris.
- Brants, T. (2000). TnT-A Statistical Part-of-Speech Tagger. In Proceedings of the 6th Applied Natural Language Processing Conference ANLP-2000 (pp. 224-231). Seattle, WA.
- Dimitrova, L., Erjavec, T., Ide, N., Kaalep, H.-J., Petkevič, V., and Tufiş, D. (1998). Multext-East: Parallel and Comparable Corpora and Lexicons for Six Central and Eastern European Languages. In COLING-ACL '98. Montreal, Quebec.
- Gale, W. and Church, K. W. (1993). A Program for Aligning Sentences in Bilingual Corpora. Computational Linguistics 19/1 (pp. 75-102).
- Di Cristo, P. (1996). MtSeg: The Multext multilingual segmenter tools. MULTEXT Deliverable MSG 1, Version 1.3.1. CNRS, Aix-en-Provence. <http://www.lpl.univ-aix.fr/projects/multext/MtSeg/>
- Danielsson, P. and Ridings, D. (1997). Practical Presentation of a "Vanilla" Aligner. TELRI Newsletter No. 5, Institute fuer Deutsche Sprache, Mannheim. <http://nl.ijs.si/telri/Vanilla/>
- Erjavec, T. and Džeroski, S. (2004). Machine Learning of Language Structure: Lemmatising Unknown Slovene Words. Applied Artificial Intelligence, 18/1 (pp. 17-41). Taylor & Francis.
- Erjavec, T. (2004). MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. Fourth International Conference on Language Resources and Evaluation, LREC'04. (pp. 1535-1538). ELRA, Paris.
- Koehn, P. (2002). Europarl: A Multilingual Corpus for Evaluation of Machine Translation. <http://people.csail.mit.edu/people/koehn/publications/europarl/>
- McEnery, T., Wilson, A., Sanchez-Leon, P., and Nieto-Serrano, A. (1997). Multilingual Resources in European Languages: Contributions of the CRATER Project. Literary and Linguistic Computing 12/4.
- Pouliquen B., Steinberger R., Ignat C. (2003). Automatic Annotation of Multilingual Text Collections with a Conceptual Thesaurus. Proceedings of the Workshop Ontologies and Information Extraction at (EUROLAN'2003). Bucharest, Romania.
- Pouliquen B., Steinberger R., Ignat C. (2004). Automatic Linking of Similar Texts Across Languages. In: Recent Advances in Natural Language Processing III. John Benjamins Publishers, Amsterdam.
- Manandhar S., Džeroski S. and Erjavec T. (1998). Learning Multilingual Morphology with CLOG. In Proceedings of Inductive Logic Programming; 8th International Workshop ILP-98 (Lecture Notes in Artificial Intelligence 1446) (pp. 135-144). Springer-Verlag, Berlin.
- Sperberg-McQueen, C. M. and Burnard, L. (eds.) (2002). Guidelines for Electronic Text Encoding and