The OPUS corpus - parallel and free http://logos.uio.no/opus/

Jörg Tiedemann[*], **Lars Nygaard**[§]

[*]Department of Linguistics and Philology
Uppsala University
Box 635, S-751 26 Uppsala, Sweden
joerg@stp.ling.uu.se

[§]Tekstlaboratoriet HF
University of Oslo
Postboks 1102 Blindern, 0317 Oslo
lars.nygaard@ilf.uio.no

Abstract

The OPUS corpus is a growing collection of translated documents collected from the internet. The current version contains about 30 million words in 60 languages. The entire corpus is sentence aligned and it also contains linguistic markup for certain languages.

1. Introduction

OPUS is a growing multilingual corpus of translated open source documents available on the Internet. The main motivation for compiling OPUS is to provide an open source parallel corpus that uses standard encoding formats including linguistic annotation. A public collection of parallel corpora that can freely be used and distributed makes it possible for everyone to run experiments on bitexts and their results can easily be compared.

2. OPUS Version 0.2

We base our corpus collection on open source documentation and their translations. Many open source projects include a large amount of textual data and invite people around the world to localise products and their documentation. Similarly to the software itself, the entire documentation is freely available and may be used in any way by anyone.

In the current version (v 0.2), the OPUS corpus includes about 30 million words in 60 languages which have been collected from three sources:

- OpenOffice.org documentation (http://www.openoffice.org)[1]
- KDE manuals including KDE system messages (http://i18n.kde.org)[2]
- PHP manuals (http://www.php.net/download-docs.php)[3]

The OpenOffice.org sub-corpus (OO) contains about 2.6 million words in six languages. The corpus is completely parallel, i.e. all English source documents have been completely translated into five languages. The KDE manual sub-corpus (KDEdoc) includes 24 languages with about 3.8 million words in total. The translation initiative at KDE is an on-going project. Hence, documents are only partly translated for many languages. New languages are added constantly. KDE system messages have been compiled into a separate sub-corpus (KDE) containing about 20 million words in 60 languages. Even this translation initiative is on-going and translations into many languages are still incomplete. The sub-corpus of PHP manuals (PHP) is derived from the HTML version of the on-line documentation of the scripting language PHP. It contains about 3.5 million words in total in 21 languages.

3. Corpus Encoding

All corpus files have been encoded in Unicode UTF8 and sentence aligned for all possible language pairs (e.g. 1830 language pairs for KDE) using a length-based approach (Gale and Church, 1993). Sentence alignments are stored in XCES format[4]. Corpus files are stored in XML using the original markup from the source documents with added linguistic markup. Additional markup includes sentence boundaries (for all documents since this is needed for sentence alignment), word boundaries (for all languages except Asian languages such as Chinese for which no tokeniser was available), part-of-speech tags (for English, French, German, Italian, and Swedish in parts of the corpus) and shallow syntactic structures (for English in parts of the corpus). We are grateful for the tools that have been provided by external researchers for adding this markup (Baldrigde, 2001; Brants, 2000; Megyesi, 2001; Matsumoto et al., 2000; Schmid, 1994).

More information will be added gradually as tools become accessible to us. Figure 1 shows an example of linguistically enriched corpus data from the OPUS corpus.

[1]OpenOffice.org is an open source office suite.

[2]The K Desktop Environment (KDE) is a free graphical desktop environment for UNIX workstations.

[3]PHP:Hypertext Preprocessor (PHP) is a widely-used general-purpose scripting language which is available as open source.

[4]XCES is the XML version of the Corpus Encoding Standard (Ide and Priest-Dorman, 2000).

```
<ul class="L2">
  <li class="">
    <p class="P4" id="8">
      <s id="s8.1">
        <chunk id="c8.1-1" type="NP">
          <w grok="NNP" tree="RB" lem="over" tnt="IN">OVER</w>
        </chunk>
      </s>
    </p>
    <p class="P5" id="9">
      <s id="s9.1">
        <chunk id="c9.1-1" type="NP">
          <w grok="NNP" tree="VBP" lem="overwrite" tnt="NNP">Overwrite</w>
          <w grok="NN" tree="NN" lem="mode" tnt="NN">mode</w>
        </chunk>
        <chunk id="c9.1-2" type="VP">
          <w grok="VBZ" tree="VBZ" lem="be" tnt="VBZ">is</w>
          <w grok="VBN" tree="VBN" lem="enable" tnt="VBN">
             enabled</w>
        </chunk>
        <w grok="." tree="SENT" lem="." tnt=".">.</w>
      </s>
```

Figure 1: Linguistic markup in OPUS. A small example from the English part of the OpenOffice.org corpus

The example in the figure is taken from the English part of the OpenOffice.org corpus which has been annotated with several linguistic tools.

First of all, a sentence splitter using simple regular expressions has been used to add sentence boundaries in form of `<s>` tags in the markup. Existing markup such as paragraph boundaries have also been used to identify sentence boundaries. Each sentence in each documents has a unique ID which is used for the sentence alignment. In the next step, a tokeniser has been used to mark words and other tokens with `<w>` tags. In the case of English we used the *Tree-Tagger* for English (Schmid, 1994) which includes a tokenisation module. This tagger produces also lemmas for each recognised word together with the part-of-speech tag. Lemma and part-of-speech are stored as attributes (`lem` and `tree`) within the XML tags for words (`<w>`). A similar tokenisation has been done for all languages for which appropriate tools have been available to us (the *Tree-Tagger* for English, French, German, Italian and *chasen* for Japanese). A simple pattern-based tokeniser has been used for other languages as default[5].

Figure 1 also illustrates how several variants of similar markup is stored in OPUS documents. The attributes `grok` and `tnt` refer to part-of-speech tags which have been assigned to each word by two other taggers, the one included in the Grok library (Baldrigde, 2001) and the TnT tagger (Brants, 2000). Including multiple versions of the same annotation type may seem redundant. However, in OPUS we try to avoid expensive manual work which is necessary for correcting errors done by automatic annotation tools. Additional information of the same annotation type can be seen as another "view" on uncertain items that can be used as an indication about the correctness of the annotation. Similar to voting techniques in classification tasks the majority of identical tags may support a certain decision. For example, the word "Overwrite" in figure 1 is twice correctly tagged as a noun and once as verb. Hence, the noun tag should be preferred.

Finally, we also applied a shallow parser to our English text documents. The parser is taken from the Grok libraries which has been trained on the Penn Tree Bank[6]. The shallow parser depends on part-of-speech tags. In our case, we decided to use the tags assigned by the Grok system as they come from the same software package. The shallow parser identifies flat chunk structures which are marked with the `<chunk>` tags in the XML document. Each chunk has a unique ID and a type attribute to describe the chunk type.

As mentioned earlier, existing markup is not removed from the original documents before any additional processing. The original markup provides valuable information such as paragraph boundaries (see `<p>` in figure 1), headers, lists (`<ul>` and `<li>` in figure 1) and tables. Maintaining this markup and the original file structure makes it possible to go back to the original source, makes it easy to produce sub-sets of the corpus, and also increases the performance of the automatic sentence alignment by reducing follow-up errors. The OPUS corpus contains a large number of fairly small documents. Aligning each pair of documents separately is much more accurate than working with large concatenated files where corrupted and incomplete translations may cause alignment errors in the following parts of the corpus. Furthermore, existing markup such as paragraph boundaries can easily be used as "hard boundaries" for synchronising the length-based alignment algorithm.

[5]The PHP corpus has not been tagged yet. It is simply tokenized using our default tokeniser for all languages.

[6]The model has been trained by Jörg Tiedemann using the maxent-module in the Grok-libraries. There is no published reference available about this model.

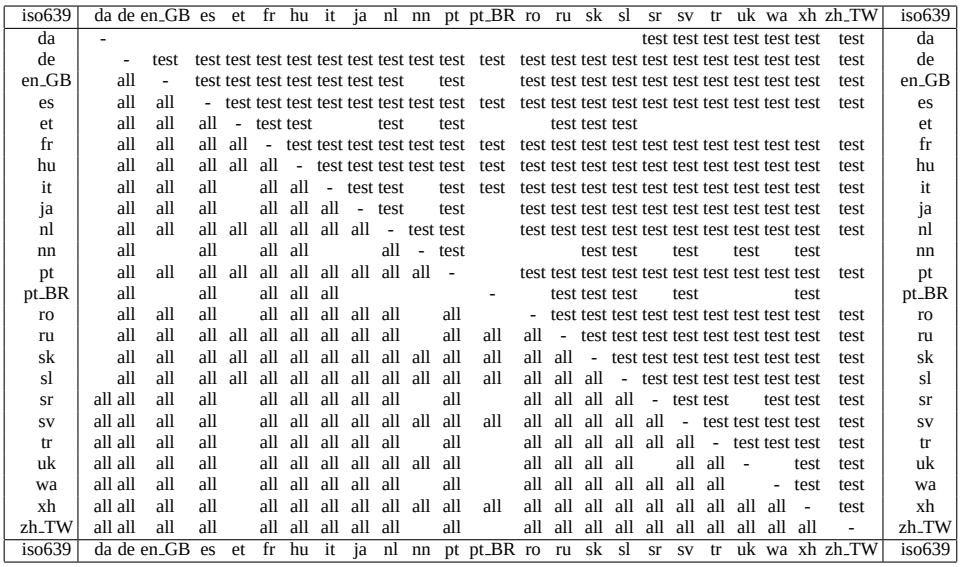

Figure 2: Available bitexts from the KDEdoc sub-corpus. Sentence-aligned HTML files can be downloaded from this matrix (complete bitexts=*all*, small sample files=*test*). Languages identifiers are coded using ISO 639-1.

4. Contents of the corpus

Using open source documentation makes it very easy to include material in a freely available corpus without having to deal with complicated copyright issues. One of the main problems with this approach is that most of the projects that supply documentation are on-going projects with partly finished material especially among manuals and their translations. OPUS is certainly not meant to serve as a representative collection of texts for the languages involved. Documents in our collection so far come from specific domains representing similar text types. Furthermore, one should not expect to find entirely correct translations of high quality in all cases. Localisation of open source software is done voluntarily by a large number of people around the world. Qualifications and skills of individuals certainly varies a lot. A quality check of the translations has not been done and is not intended to be done within the project. Furthermore, the amount of translated material also differs a lot. These drawbacks have to be taken into account when working with the material in OPUS.

The main benefits (besides being free of charge) include the variety of languages in the corpus and the alignment between all possible language pairs in the corpus. These features make OPUS a unique source of linguistic data for all kinds of investigations. Translation studies and other linguistic examinations are possible as well as machine learning techniques for the acquisition of linguistic parameters in natural language processing. The specificity of the corpus may also invite studies on certain sub-languages and their translations to other languages.

5. Tools

Within the project several tools have been used for conversion, annotation and data management. We do not provide any tools for the time being but we plan to include a collection of scripts and tools within OPUS in the future. They will include tools for collecting data from the web, for corpus annotation, and for processing files in the corpus. There are some search facilities freely accessible from the project home page. We plan to add further on-line tools for larger parts of the corpus. We also work on tools for the automatic retrieval of translation data from the web and for the extraction of multi-lingual terminology from the corpus. Tools and term databases will be available from the project.

6. Availability

The entire corpus is freely available from the project homepage (http://logos.uio.no/opus/). It can be downloaded in its native XML source format or compiled as sentence aligned HTML-documents (see figure 2). Parts of the corpus are accessible via on-line search facilities. Multi-lingual concordancers using the Corpus Work Bench (Christ, 1994) are available for the OpenOffice.org subcorpus (except the Japanese translation). The result of an example query is shown in figure 3.

We are currently working on adding further material. The contents of the collection will be updated continuously, and updates will be announced on the project web page. The next release will include the EUROPARL corpus[7] encoded and annotated in the OPUS style. We also work with additional on-line documentations and multilingual news which will be added to the corpus soon. Furthermore, we will extend the on-line query system to include larger parts of the corpus in the near future.

7. Feedback and contributions

The main advantage of open source projects is the possibility of unrestricted co-operations between interested people around the world. Similarly to other open source initiatives we invite everybody to contribute to the project. We

[7]EUROPARL includes proceedings of the European Parliament in 11 languages (Koehn, 2003). It is provided by Phillipp Koehn from http://www.isi.edu/~koehn/europarl/

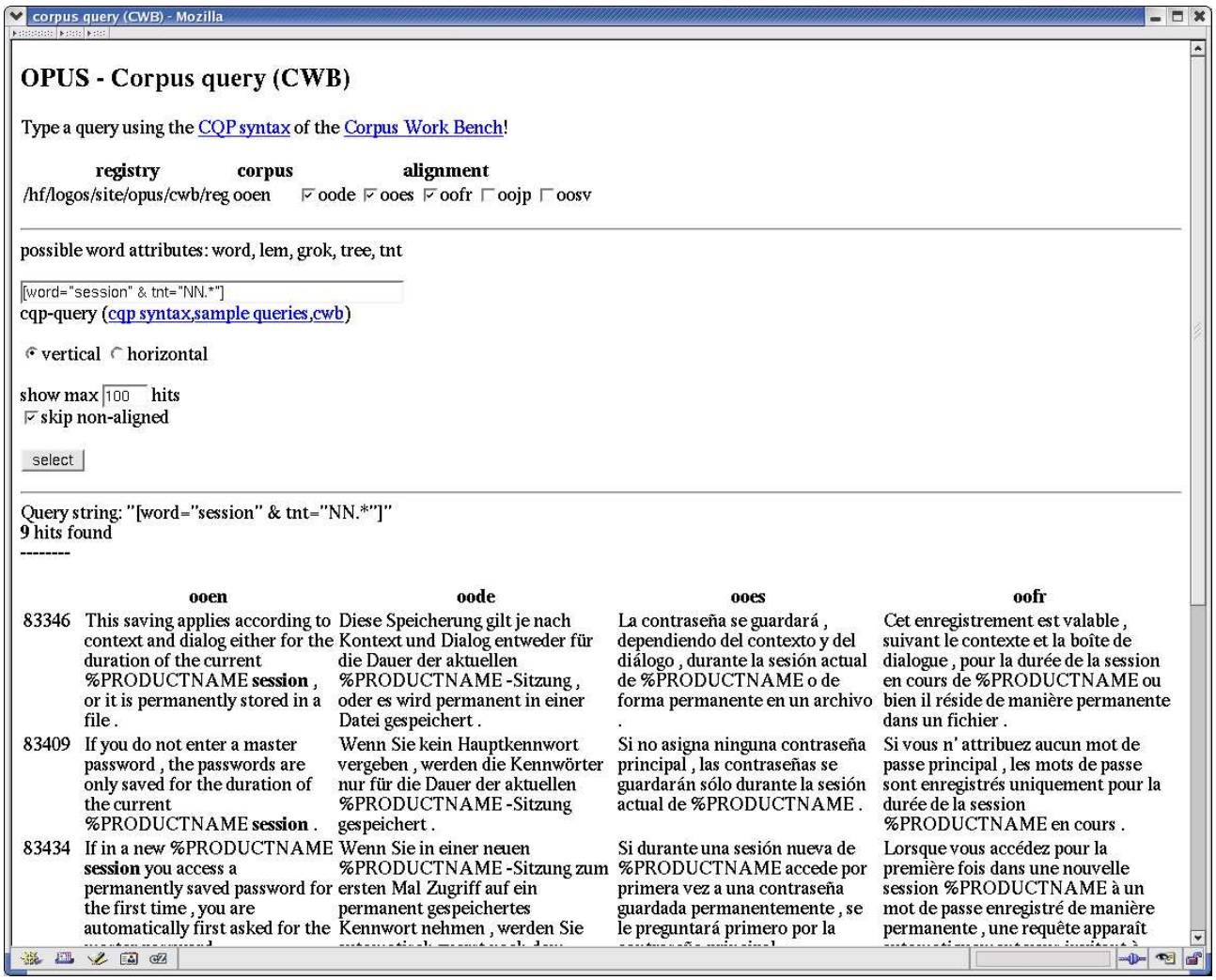

Figure 3: A multilingual concordance tool for the OpenOffice.org corpus.

would like to collect suggestions and other kinds of feedback for improving OPUS and its contents. We would also like to ask for contributions in terms of data, tools and quality control. Please let us know if you use any part of OPUS and what your experiences were with the data. Feel free to contact the members of the project.

8. References

Baldrigde, Jason, 2001. Grok - an open source natural language processing library.

Brants, Thorsten, 2000. TnT - a statistical part-of-speech tagger. In *Proceedings of the 6th Applied Natural Language Processing Conference ANLP-2000*. Seattle, WA.

Christ, Oliver, 1994. A modular and flexible architecture for an integrated corpus query system. In *Proceedings of the 3rd International Conference on Computational Lexicography (COMPLEX)*. Budapest.

Gale, William A. and Kenneth W. Church, 1993. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102.

Ide, Nancy and Greg Priest-Dorman, 2000. Corpus encoding standard - document CES 1. Technical report, Department of Computer Science, Vassar College, and Equipe Langue et Dialogue, New York, USA and LORIA/CNRS, Vandoeuvre-lés-Nancy, France.

Koehn, Philipp, 2003. Europarl: A multilingual corpus for evaluation of machine translation. Unpublished draft, available from `http://www.isi.edu/~koehn/europarl/`.

Matsumoto, Yuji, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Hiroshi Matsuda, Kazuma Takaoka, and Masayuki Asahara, 2000. Morphological analysis system chasen version 2.2.1 manual. http://chasen.aist-nara.ac.jp/chasen/bib.html.en.

Megyesi, Beáta, 2001. Comparing data-driven learning algorithms for POS tagging of swedish. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Carnegie Mellon University, Pittsburgh, PA, USA.

Schmid, Helmut, 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*. Manchester, UK. Http://www.ims.uni-stuttgart.de/~schmid/.