CONTENTS

News from OPUS — A Collection of Multilingual Parallel Corpora with Tools and Interfaces

Jörg Tiedemann

University of Groningen

Abstract

The OPUS corpus is a growing resource providing various multilingual parallel corpora from different domains. In this article we introduce resources that have recently been added to OPUS. We also look at some corpus-specific problems and the solutions used in preparing the parallel data for the inclusion in our collection. In particular, we discuss the alignment of movie subtitles and the conversion of biomedical documents and localization data to a sentence aligned XML format. OPUS also includes various tools and interfaces besides the actual data. We will briefly describe our corpus processing and query tools and a newly added lexical database of word alignments.

1 Introduction

Parallel corpora are essential resources for a wide range of applications in natural language processing and corpus linguistics. The interest in parallel data has grown dramatically especially due to the boom in research on statistical machine translation (SMT) in the recent years. OPUS (Tiedemann & Nygaard 2004) tries to provide the research community with a wide range of freely available parallel corpora in many languages. The main focus is to collect parallel documents from various domains and to pre-process them in such a way that they are directly useful for applications such as statistical machine translation and multilingual terminology extraction. We emphasize the inclusion of a large number of languages in order to support under-resourced languages. Multilingual data is taken from several on-line sources. All documents are converted to a uniform XML format and all possible language pairs are aligned at the sentence level. Initially, OPUS included localization data and manuals of open-source software (Tiedemann & Nygaard 2004). Thereafter, political and administrative texts from the European Union have been added and converted to the OPUS format. Recently, a large database of movie subtitles in many languages has been added and the latest sub-corpus comes from yet another domain: biomedical data from the European Medicines Agency (EMEA). In addition to new data collections, we also extend the annotation of existing data sets. For example, large portions of the Dutch corpora have automatically been parsed using the wide-coverage dependency parser Alpino (van Noord 2006) and the

machine-annotated treebanks are available on-line. Finally, OPUS also provides interfaces for querying the corpus data. Several interfaces are available for searching through the parallel data. The latest addition is a database with an on-line interface that provides multilingual lexical word type links derived from automatic word alignment. Details of all these extensions are discussed below.

2 Recently added corpora

In this section we describe recently added corpora in the OPUS collection. We include some discussion of corpus-specific pre-processing and alignment issues.

2.1 *OpenSubtitles — A parallel corpus of movie subtitles*

There are several on-line databases providing movie subtitles in various languages. They usually collect user uploads that can be searched in various ways. A very reliable source is OpenSubtitles.org, which offers an extensive multilingual collection of movie subtitles. The providers of this website were very co-operative and gave as their entire database of about 308,000 files covering about 18,900 movies in 59 languages (status of July, 2006).

2.1.1 *Pre-processing*

Unfortunately, the subtitle collection includes a lot of noise, which, of course, is to be expected from an open database of user uploads. Several pre-processing and filtering steps had to be taken in order to clean up the collection at least to a reasonable extent. Subtitles are provided in various textual formats and characters encodings. We decided to use the so-called "subviewer" format (usually with the extension `.srt`). Files in another popular format, the microDVD format (with extension `.sub` have been converted to subviewer format using a freely available script sub2srt (Obermayer, 2005). Other files have been discarded.

All files are then converted to Unix format, Unicode UTF-8 encoding and checked by a language guesser using `textcat` (van Noord, 1997), which we trained for 46 languages. Files that have been tagged with the same language as being guessed by `textcat` have been selected to be included in our corpus. In this way we removed a lot of garbage from the database including subtitle files with corrupted contents, wrong language tags and incorrect or unknown character encoding. Unfortunately, we also loose a lot of valuable data by excluding languages for which no language model has been trained. However, we still store these files in a separate folder (called 'unknown'). We plan to provide even those files in future releases.

Furthermore, in various cases `textcat` provides several possible labels. Also these cases are discarded to yield the highest precision in our selection. We keep subtitles for which the first one of the guessed languages corresponds to the labeled language are stored in a folder called 'maybe' and the ones for which one of the guessed languages corresponds to the labels are stored in 'probably not'. We will also make them available in future releases. Finally, there are a lot of copies in the database due to multiple uploads of subtitles for the same movie. For those only the latest one is used in OPUS assuming that a new upload is mainly done in order to correct an erroneous previous one. However, we include multiple copies of subtitles for the same movie if they correspond to different video files and have a corresponding subtitle file in a different language in the database.

The last step in pre-processing includes the conversion to XML as used in OPUS. This includes sentence splitting and tokenization. We developed a simple script doing this conversion. Sentence splitting and tokenization is basically been done using regular expressions tailored towards subtitle data. Language specific treatment is still very limited. Specific tokenization procedures have been included for Chinese (using a lexicon based segmenter (Peterson; thanks to Yan Zhao for providing the lexical data), for Japanese (using ChaSen (Matsumoto & Kitauchi)), and for Dutch (using the Alpino tokenizer (van Noord 2006)).

After pre-processing and language checking we retained 38,825 subtitle files in 29 languages. From that we selected 22,794 pairs of subtitles for alignment (selecting only the ones corresponding to the same physical video file) covering 2,780 movies in 361 language pairs. Altogether, this corresponds to about 22 million sentence alignments created by the approach described below.

2.1.2 *Sentence alignment*

As already discussed in previous papers (Tiedemann 2007, 2008) and also described in related studies (Itamar & Itai 2008, Armstrong et al. 2006), traditional sentence alignment approaches are not appropriate for the alignment of movie subtitles. An obvious idea is to use the time information from subtitle files for the alignment. For our corpus we applied such an approach entirely based on the timing information (Tiedemann 2007, 2008). Another approach would be to combine various types of information for the alignment (see, for instance, (Itamar & Itai 2008) for a combination of length and time information).

There are several problems with a time-based alignment approach. Firstly, in our corpus we work with the alignment of actual sentences (as opposed to subtitle frame alignment). This means that sentence may span several

time slots or may start or end within a time slot. This problem can be solved by interpolating the time given in the subtitle files to the points of sentence boundaries. We used a simple linear interpolation based on the ratio of string length and time which seems to work sufficiently well. Secondly, time information is unfortunately not very reliable. There are often slight differences in the timing that cause devastating errors when aligning purely based on this information. Solving this problem basically requires a synchronization of both subtitle files. Fortunately, the time differences seem to be very consistent depending on only two parameters, time offset and speed difference (which we will call *time ratio*). Both parameters can simply be calculated using two fixed anchor points of true correspondence using the formulas given below.

$$\begin{aligned} time_{ratio} &= \frac{(trg_1 - trg_2)}{(src_1 - src_2)} \\ time_{offset} &= trg_2 - src_2 * time_{ratio} \end{aligned}$$

Here, src_1 and src_2 correspond to the time values (in seconds) of the anchor points in the source language and trg_1 and trg_2 to the time values of corresponding points in the target language. Using $time_{ratio}$ and $time_{offset}$ we then adjust all time values in the source language file before aligning them using our time overlap approach.

The time synchronization approach described above is very effective and yields significant improvements where timing differences occur. However, it requires two reliable anchor points that should also be far away from each other to produce accurate parameter estimations. In order to reduce manual intervention we use the following heuristics to select appropriate anchor points automatically. Firstly, we search for cognates in the beginning and at the end of each subtitle pair using sliding windows and a fixed number of sentences. For this, we use the longest common subsequence ratio with a fixed score threshold. This is quite effective for language pairs that use the same alphabet. Also less related language pairs can be processed in this way because subtitles often include many names which are often good candidates for synchronization. Clearly, the cognate approach has its limitations especially for language pairs with different alphabets. Therefor we add a second strategy based on bilingual dictionaries. Anchor point candidates are then searched in the same fashion using sliding windows but using dictionary entries for matching. In order to keep the approach independent of language resources, we applied automatic word alignment to create rough bilingual dictionaries from the data itself. In other words, we align all subtitles without the dictionary-based synchronization on the sentence level and run GIZA++ (Och & Ney 2003) on this data to create

alignments between words. We use some heuristics and filtering techniques to increase the precision of the alignment and extract word type links from the bitexts. In this way, we expect to obtain rough bilingual dictionaries even from imperfectly aligned resources assuming that spurious alignments are not very consistent and, therefore, fall out after filtering. For more details about this approach, see (Tiedemann 2008).

A last decision that has to be made is the selection of the most appropriate synchronization points from the candidates obtained using the techniques described above. For this we apply another heuristics, assuming that good sentence alignment includes only a few empty links, i.e. insertions or deletions of sentences. Therefore, we define the alignment type ratio as follows:

$$algtype_{ratio} = \frac{|\text{non-empty links}| + 1}{|\text{empty links}| + 1}$$

Using the ratio above as an indicator for alignment quality, we can now test all possible pairs of anchor point candidates and measure their appropriateness in terms of synchronization. Fortunately, time-based alignment is fast enough to enable a extensive search for the best setting according to the alignment type ratio.

Testing various approaches with about 1000 reference alignments from 10 randomly selected movies yields the following results (see table 1).

Dutch - English			
approach	correct	partial	wrong
length	0.397	0.095	0.508
time	0.599	0.119	0.282
time-cog	**0.738**	0.115	0.147
time-dic	**0.765**	0.131	0.104

Dutch - German			
approach	correct	partial	wrong
length	0.631	0.148	0.220
time	0.515	0.085	0.400
time-cog	**0.733**	0.163	0.104
time-dic	**0.752**	0.148	0.100

Table 1: *The quality of different alignment approaches:* length *refers to the baseline using a length-based alignment approach,* time *refers to the time-slot overlap approach. The extension* cog *refers to the application of the cognate filter and* dic *to the dictionary approach.*

The dictionary-based method (time-dic) clearly outperforms the other sentence alignment approaches for both language pairs tested. In our final setting, we used a combination of the cognate and the lexicon based synchronization techniques. However, we did not evaluate the results but we expect at least similar results as the highest scoring one. The aligned subtitle corpus is available from the OPUS website `http://www.let.rug.nl/~tiedeman/OPUS/OpenSubtitles.php`.

2.2 EMEA — *a corpus of biomedical documents*

A recent addition to OPUS includes biomedical data retrieved from the European Medicines Agency (EMEA). The corpus includes documents related to medicinal products and their translations into 22 official languages of the European Union. It contains roughly 1,500 documents for most of the languages; not all of them are available in every language. The data has been processed in a similar way as other corpora in OPUS. In particular, the entire corpus has been converted into XML and all language pairs have been sentence aligned. It comprises 231 bitexts with a total of more than 22 million sentence fragments. The sizes of the bitexts vary between 700,000 and 900,000 aligned units. Table 2 includes some statistics of the corpus.

lang	files	tokens	sentences	lang	files	tokens	sentences
bg	1,117	11,748,464	834,711	it	1,628	13,445,886	970,921
cs	1,565	11,707,485	940,489	lt	1,563	11,045,474	957,885
da	1,634	12,156,840	1,022,499	lv	1,567	11,109,658	941,351
de	1,652	12,059,895	1,066,994	mt	988	12,316,401	776,762
el	1,632	13,731,478	1,016,148	nl	1,628	12,503,233	981,669
en	6,591	30,580,774	2,143,022	pl	1,571	12,230,972	959,959
es	1,667	13,818,929	998,015	pt	1,631	13,828,388	979,810
et	1,569	10,178,389	936,264	ro	1,109	11,914,802	851,219
fi	1,627	10,472,772	998,184	sk	1,569	11,633,259	942,550
fr	1,645	14,513,025	996,904	sl	1,567	12,128,757	945,213
hu	1,564	11,630,737	965,739	sv	1,625	11,535,592	981,738

Table 2: *The size of the* EMEA *corpus per language*

The contents of the EMEA corpus is very domain specific containing specialized terminology and repeated expressions. Therefore, this corpus can be seen as an interesting resource for building a strictly domain specific application and for investigating its specialized terminology and linguistic structures.

2.2.1 *Pre-processing*

The EMEA corpus has been compiled out of PDF documents available online. After downloading these documents they first had to be converted to plain text format, which was done using the freely available tool `pdftotext` from the `xpdf` package. The tool is quite robust and supports several text encodings such as KOI8-R (Cyrillic), IOS-8859-2 (Latin 2 for Eastern European Languages), ISO-8859-7 (for Greek) and ISO-8859-8, ISO-8859-9 (for Hebrew and Turkish, which are not used in EMEA). We also used the '-layout' option to maintain the physical layout of the document as much as possible. After some experimentation we concluded that layout

information was very important for subsequent pre-processing steps such as sentence splitting and tokenization. However, this caused problems with structures such as columns and tables. For handling them, we applied a simple post-processing script implementing some heuristics for the conversion of columns and tables into running text. Basically, the script looks through the output of `pdftotext` and checks if subsequent lines have text starting at identical positions, which often indicates column structures. This strategy is complicated by the fact that not all table cells or columns have to be filled with text at each line. Therefore, we used an approach that checks the compatibility of possible column structures in the following way. First we try to detect the start of a column or table structure using the heuristics that columns or table cells should be separated by at least three space characters. After detecting such a line we add subsequent lines if they do not violate the anticipated structure. Violations are caused by text running over column/cell boundaries. The first non-compatible line ends the section and columns/tables are converted to running text — one column/table-cell after another.

After this conversion all text files are tokenized and stored in the OPUS typical XML format. Some language specific tools are used to improve tokenization, sentence splitting and to add additional annotation such as POS tags and chunk labels.

2.2.2 *Sentence alignment*

In the pre-processing step XML document have been created, which can be used by the Uplug tools with its integrated sentence aligners (Tiedemann). They are sorted into language specific sub-directories and sentence alignment is then performed for all corresponding files (determined by their file names). We used 'hunalign' (Varga et al. 2007) with the 'realign' feature for this purpose which seems to produce very reliable results according to our experience. However, we did not measure the quality of the automatic alignment explicitly. The sentence links are stored in external files as in all other OPUS corpora and, therefore, corrections can easily be made or other types of automatic alignment can be performed.

The EMEA corpus including all sentence alignments are available from `http://www.let.rug.nl/~tiedeman/OPUS/EMEA.php`. There are also plain text files available for each bitext besides the XML based representation. The Dutch portion has also been parsed by Alpino and the treebank is on-line as well.

2.3 KDE4 — *Localization data in many languages*

The last new data collection to be presented here is an extension of a resource already previously used in OPUS — the localization files of KDE. We

downloaded the latest set of localization files for KDE version 4 and converted them into a parallel corpus. KDE supports more than 80 languages. However, not all translations are completed and, therefore, the KDE4 corpus is not entirely parallel. Localization files are available in a simple format using unique message IDs (msgid) to identify a message and message strings (msgstr) to store the translation string. The message ID is usually the original message in English that corresponds to the translated string to be shown in the localized version.

We used a simple script to convert these localization files into aligned XML files. This script merges multi-line messages, removes hotkey markers ('&') and checks HTML style markup. It adds some basic XML markup including a header with meta information extracted from the localization files. The XML documents are then checked (and corrected if necessary) in a post-processing step the tool "tidy" (Ragett).

For simplicity we did not perform any further sentence splitting but left each message ID and its translation as one textual unit to be aligned (treating each message as one single sentence). In this way we get highly accurate alignments, but the sentence markup is not optimal as some messages contain more than one sentence. However, most messages are very short and mainly consist of only one sentence or just a term or phrase. Still, the sentence splitting problem should be address in a future release. Note that we use the message ID not only for aligning all languages to English but also to align every other language pair. The English message ID is then used as a unique anchor to link various translations together. In this way, we obtain a large number of bitexts from the localization files,

Finally we add further annotation for some languages. Here we use the same tools as for other OPUS corpora including POS taggers and chunkers. The KDE4 corpus with all its bitexts is available from
`http://www.let.rug.nl/~tiedeman/OPUS/KDE4.php`.

3 Tools and interfaces

OPUS is not only a collection of data but also includes various tools and interfaces to process and browse through the corpora provided. Here we give a brief overview of some of the tools available.

3.1 *Corpus processing tools*

We have used various tools for preparing the corpora included in OPUS. The main strategy is to re-use available resources as much as possible and to always apply annotation tools that we have at our disposal. A non-comprehensive list of tools is available at the OPUS website

http://www.let.rug.nl/~tiedeman/OPUS/tools.php. In particular, we apply various types of open-source software and free research tools. Many OPUS specific tools have been integrated in Uplug (Tiedemann) which is intensively used for producing the corpus files. They are freely available and can also be used by others to produce similar corpora. Corpus specific tools have been developed to, for example, convert and align movie subtitles (srt2xml.pl & srtalign.pl). They are also available via Uplug and the OPUS web site and can easily be used for producing parallel subtitle corpora.

Furthermore, we provide tools for browsing through and converting OPUS corpora. In particular, there is a simple script for browsing through sentence aligned bitexts by converting the XML and the external sentence alignments to plain text format. There is also a similar script that allows the conversion to the popular Moses/GIZA++ format that is used in training statistical machine translation systems. Here, additional annotation such as POS tags can also be used to create input files with various factors.

The last tool to be mentioned here is related to the query interfaces described below. For querying our corpora we use the Corpus Work Bench originally developed by IMS Stuttgart (Christ 1994). We implemented a tool that converts OPUS data to the input formats necessary for indexing parallel corpora with CWB and which calls appropriate programs to create the internal structures. It automatically supports indexing of the additional linguistic annotation included in many OPUS corpora. We used the script extensively to create query databases for all parallel corpora in OPUS.

3.2 *Multilingual corpus query interfaces*

There are basically two types of interfaces for querying OPUS corpora via the corpus work bench (CWB). One interface can be used as a general query engine for all corpora included in OPUS. It is available at http://www.let.rug.nl/~tiedeman/OPUS/bin/opuscqp.pl and supports queries for any combination of parallel data available using the CWB query language (CQP syntax). The output can be formatted in different ways (KWIC format, horizontal alignment, vertical alignment) and may include additional annotation such as POS tags if available for the particular corpus.

For some parallel corpora a second type of interface is available. This is essentially based on the example provided by the CWB package with support for aligned corpora added. These interfaces support various kinds of highlighting and display styles depending on the annotation available (for example bracketing with labels if chunk information is annotated). Query results are cached for faster access and may be browsed page by page. Additional context may also be shown. Furthermore, they include additional features such as frequency counts. Currently, we have this type of interface

available for the Europarl corpus, the OpenSubtitle corpus and the corpus of the (dismissed) Constitution of the EU.

3.3 *The word alignment database*

Recently, a word alignment database has been added to the OPUS repository. Here, we collect word type links derived from automatic word alignment using GIZA++. We used the standard alignment models that are implemented in that system yielding directional links between words in parallel corpora. It assigns one link per target language word and, hence, does not allow n:m alignments. Several heuristics exist to combine directional alignments (source-to-target and target-to-source) in order to "symmetrize" word alignment results. In our task, bilingual lexicon extraction, we focus on precision rather than recall and, hence, we like to focus on the most reliable links. Therefore, we used the intersection of directional alignments which is know to produce the most confident links between words in source and target language. However, a disadvantage of this approach is the fact that this heuristics only allows for one-to-one word links which in many cases is not satisfactory. Therefore, we also computed an alignment combination known as "refined" which incrementally adds adjacent links to the intersection of links in order to form n:m alignments. More details about these heuristics can be found in Och & Ney (2003) and Tiedemann (2004). In order to improve precision we applied some further filtering after extracting word type links from word aligned parallel corpora. Firstly, we selected links with an alignment frequency of 5 or more. Secondly, we restricted ourselves to lexical items which include alphabetical characters only. In this way, we obtain lists of word type pairs with high confidence sorted by alignment frequency. These lists have been generated for all language pairs for three of the sub-corpora in OPUS: Europarl, EUconst and OpenSubtitles. They are accessible in a multilingual database via an on-line web-interface (`http://urd.let.rug.nl/tiedeman/OPUS/lex.php`). A screen shot is shown in figure 1.

The database and its interface include additional features such as user feedback (judging the correctness of a link), sub-corpus selection, and a connection to the bilingual concordance tool showing examples of aligned sentences containing the selected words. The database currently includes 31 languages. We hope to extend it in the near future with additional word pairs and languages coming from other sub-corpora in OPUS.

4 Conclusions

In this article we presented various recent extensions of OPUS. In particular, we described two additional parallel corpora included in our collection:

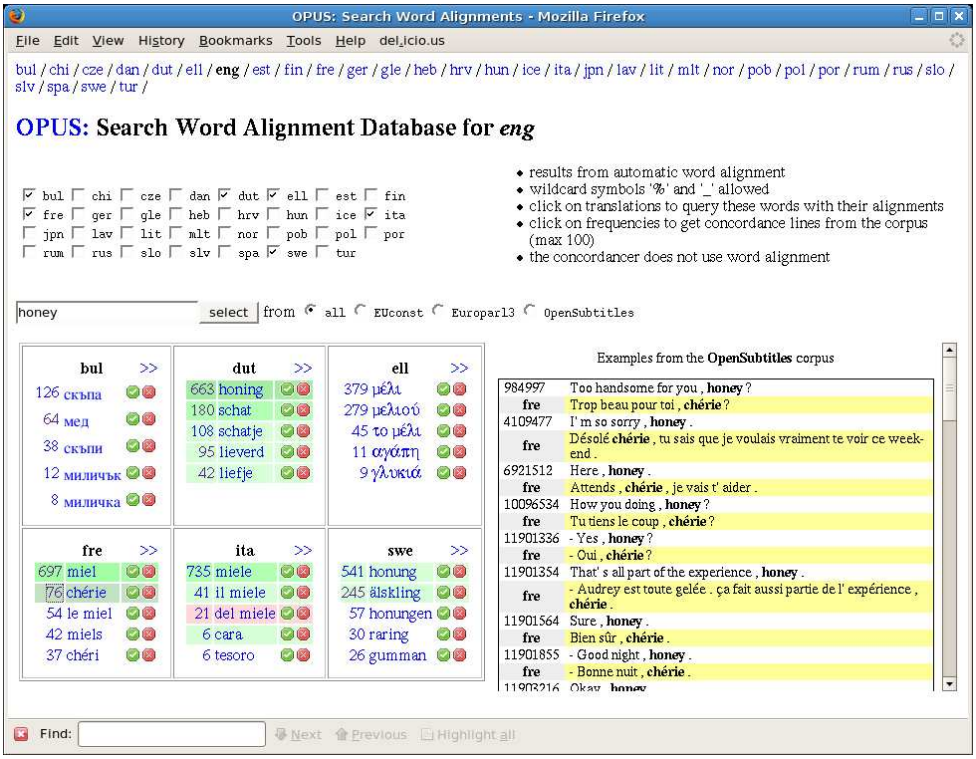

Fig. 1: *The* OPUS *Word Alignment Database: Alignments of the English word "honey" with color-coded user feedback and English-French concordance lines to the left*

an extensive parallel corpus of biomedical documents and a corpus of the latest edition of localization files of KDE4. We also discussed improvements of domain-specific sentence alignment applied to the OpenSubtitles corpus. Including cognates and automatically created word type dictionaries helps to synchronize movie subtitles for better matching of their timing information. Finally, we also described tools and on-line interfaces for accessing our data collections. Among others, we provide a word alignment database with user feedback functionality in an on-line interface.

REFERENCES

Armstrong, S., C. Caffrey, M. Flanagan, D. Kenny, M. O'Hagan & A. Way. 2006. "Leading by Example: Automatic Translation of Subtitles via EBMT". *Perspectives: Studies in Translatology* 14:3.163-184.

Christ, O. 1994. "A Modular and Flexible Architecture for an Integrated Corpus Query System". *Proceedings of the 3rd International Conference on Compu-*

tational Lexicography (*COMPLEX*), 22-32. Budapest.

foolabs. Xpdf – A Toolkit for Viewing and Processing PDF Documents. `http://www.foolabs.com/xpdf/` [Source checked in Oct. 2007]

Itamar, E. & A. Itai. 2008. "Using Movie Subtitles for Creating a Large-scale Bilingual Corpora". *6th Int. Conf. on Language Resources and Evaluation* (*LREC'08*). Marrakech, Morocco.

Matsumoto, Y. & A. Kitauchi. 2007. Chasen – A Japanese Morphological Analysis System, version 2.2.9. `http://chasen.naist.jp/hiki/ChaSen/` [Source checked in Oct. 2007]

Obermayer, R. 2005. sub2srt – A Tool for Converting Subtitles from .sub to .srt Format, version 0.5.3. `http://www.robelix.com/sub2srt/` [Source checked in Oct. 2007]

Och, F.J. & H. Ney. 2003. "A Systematic Comparison of Various Statistical Alignment Models". *Computational Linguistics* 29:1.19-51.

OpenSubtitles.org – A Repository of Subtitles. `http://www.opensubtitles.org` [Source checked in Oct. 2007]

Peterson, E. 2007. A Segmentation Tool for Chinese. `http://www.mandarintools.com/segmenter.html` [Source checked in Oct. 2007]

Raggett, D. 2003. Clean up Your Web Pages with HTML Tidy. `http://www.w3.org/People/Raggett/tidy/`. [Source checked in Oct. 2007]

Tiedemann, J. 2003, *Recycling Translations - Extraction of Lexical Data from Parallel Corpora and their Application in Natural Language Processing.* Doctoral Thesis, Studia Linguistica Upsaliensia 1. `http://sourceforge.net/projects/uplug`. [Source checked in Oct. 2007]

Tiedemann, J. 2007. "Improved Sentence Alignment for Movie Subtitles". *Int. Conf. on Recent Advances in Natural Language Processing* (*RANLP 2007*), 582-588. Borovets, Bulgaria.

Tiedemann, J. 2008. "Synchronizing Translated Movie Subtitles". *6th Int. Conf. on Language Resources and Evaluation* (*LREC'08*). Marrakech, Morocco.

Tiedemann, J. & L. Nygaard. 2004. "The OPUS Corpus – Parallel and Free". *4th Int. Conf. on Language Resources and Evaluation* (*LREC'2004*), 1183-1186. Lisbon, Portugal.

van Noord, G. 1997. "TextCat – An Implementation of a Text Categorization Algorithm". `http://www.let.rug.nl/ vannoord/TextCat/` [Source checked in Oct. 2007]

van Noord, G. 2006. "**A**t **L**ast **P**arsing **I**s **N**ow **O**perational". *TALN 2006 Verbum Ex Machina, Actes De La 13e Conference sur Le Traitement Automatique des Langues Naturelles*, 20-42, Leuven, Belgium.

Varga, D., P. Halácsy, A. Kornai, V. Nagy, L. Németh & Viktor Trón. 2007. "Parallel Corpora for Medium Density Languages". *Recent Advances in Natural Language Processing IV* (= *Current Issues in Linguistic Theory*, 292) ed. by N. Nicolov et al., 247-258. Amsterdam & Philadelphia: John Benjamins.

Index of Subjects and Terms