SEVENTH FRAMEWORK PROGRAMME
THEME 3
Information and communication Technologies

PANACEA Project

Grant Agreement no.: 248064

Platform for **A**utomatic, **N**ormalized **A**nnotation and
Cost-**E**ffective **A**cquisition
of Language Resources for Human Language Technologies

D5.2

Aligners Integrated Into The Platform

Dissemination Level:	Public
Delivery Date:	January 31st 2011
Status – Version:	Final v1.0
Author(s) and Affiliation:	Antonio Toral (DCU), Pavel Pecina (DCU), Andy Way (DCU)

PANACEA is a EC funded project under Grant Agreement 248064

This document is part of technical documentation generated in the PANACEA Project, **P**latform for **A**utomatic, **N**ormalized **A**nnotation and **C**ost-**E**ffective **A**cquisition (Grant Agreement no. 248064).

Please send feedback and questions on this document to: iulatrl@upf.edu

TRL Group (Tecnologies dels Recursos Lingüístics), Institut Universitari de Lingüística Aplicada, Universitat Pompeu Fabra (IULA-UPF)

Relevant Documents

D3.1 Requirement analysis of the platform.
D5.1 Report describing the technologies and tools for the creation and alignment of parallel corpus.
D3.2 First version of the integrated platform and documentation.
D4.2 Initial functional prototype and documentation describing the initial CAA subsystem and its components.

Table of Contents

1 Introduction

This document constitutes the second milestone of work-package 5. It reports on the integration into the Panacea platform of the aligners described in Deliverable 5.1 (Section 5 and appendix A).

The rest of the document is structured as follows. The next section details the implementation of webservices for each of the aligners considered. Next, we discuss on the procedures developed to convert from the format of these aligners to the Travelling Object format. Subsequently, we present a set of workflows that demonstrate the usage of aligners in acquisition pipelines. After that, we report on the software that has been developed, which is attached to the deliverable. Finally, we sum up the work carried out, and its role in forthcoming tasks of the project.

2 Webservices

Webservices created for sentential aligners are covered in section 2.1 while section 2.2 deals with sub-sentential aligners. The work done covers several points of the to-do list introduced in D3.1 (section 7.1), namely: TO-GR-02 (conversion to simple TO format), TL-W5-01 (WP5 aligners) and WS-CI-01 (common interfaces). All of the following webservices can be accessed at http://www.cngl.ie/panacea-soaplab2-axis/.

2.1 Sentential alignment

2.1.1 Hunalign

Hunalign (Varga et al., 2005)[1] takes three mandatory parameters (source corpus, target corpus and bilingual dictionary, although the last one can be empty). The first two are mandatory in the webservice, together with the source and target languages. The bilingual dictionary file is optional, if none is provided the webservice will create and use an empty file.

Hunalign provides a set of optional parameters too. Some of them are offered by the webservice (bisent, cautious and text), while two of them are activated internally (realign and utf). The remaining ones regard evaluation purposes or postfiltering and have been discarded.

2.1.2 GMA

GMA – Geometric Mapping and Alignment (Argyle et al., 2004)[2] – takes a pair of parameters for the source and target corpus. Apart from that, it needs a parameter pointing to a configuration file, which contains several parameters, including language-dependent lists of stop words. The webservice offers two parameters for the source and target languages; these denote a language pair, which is assigned internally a configuration file. GMA provides configuration files and stop words for English-French. Additional configuration files have been created for the following

1 http://mokk.bme.hu/resources/hunalign
2 http://nlp.cs.nyu.edu/GMA/

language pairs: English-German, English-Greek, English-Spanish and English-Italian (all the parameters have the same values across language pairs except for those that are language-dependent). The stop word lists used for English, French, German, Spanish and Italian have been obtained from Université de Neuchâtel[3] while the one for Greek has been provided by ILSP.

2.1.3 BSA

BSA – Bilingual Sentence Aligner (Moore, 2002)[4] – takes three parameters (source corpus, target corpus and a threshold). All of them are offered by the webservice developed. The threshold is set to 0.5 by default. The output of this tool was altered by modifying the script "filter-final-aligned-sents.pl". This script carries out the last phase of the alignment; it receives the set of word alignments found and performs some filtering outputting only those that have high probability. BSA originally outputs a couple of files with the aligned text (one sentence per line). Conversely, we are interested in a unique file where each line contains an alignment by giving the sentence numbers of the source and target languages. The output has been modified to provide a tabbed format following the format provided by Hunalign.

2.2 Sub-sentential alignment

2.2.1 GIZA++

GIZA++ (Och and Ney, 2003)[5] performs word alignment in several steps that involve different tools (plain2snt, mkcls, snt2cooc, GIZA++, giza2bal and symal). The webservice developed encapsulates all of them through a unique call.

This word aligner toolkit offers many fine-grained input options, which are mainly numeric parameters that modify the behaviour of the aligner. Being the emphasis of the platform to provide easy-to-use versions of the tools, most of the parameters have been kept fixed (i.e. they cannot be changed through the webservice interface). These values (taken from the Moses Machine Translation system[6]) are,

- For GIZA++: -model1iterations 5, -model2iterations 0, -model3iterations 3, -model4iterations 3, -model1dumpfrequency 1, -model4smoothfactor 0.4, -nodumps 1, -nsmooth 4, -onlyaldumps 1, and -p0 (parameter p_0 in IBM-3/4) 0.999
- For symal: -alignment grow, -diagonal yes, -final yes and -both yes

The parameters offered are source and target languages (2-character ISO codes) and the source and target corpora. Apart from these general parameters, the webservice interface offers two specific parameters for mkcls: number of iterations (default value 2) and number of classes (default value 50).

3 http://members.unine.ch/jacques.savoy/clef/index.html
4 http://research.microsoft.com/en-us/downloads/aafd5dcf-4dcc-49b2-8a22-f7055113e656/
5 http://code.google.com/p/giza-pp/
6 http://www.statmt.org/moses/

2.2.2 BerkeleyAligner

BerkeleyAligner (Haghighi et al., 2009; DeNero and Klein, 2007; Liang et al., 2006),[7] similarly to GIZA++, provides a plethora of options (see documentation/manual.txt in its distribution for details). Most of them are not considered in the webservice interface. The only parameters that are offered by the webservice are the following: source and target corpora, source and target languages (2-character ISO codes) and number of iterations to run the model (default value 2).

2.2.3 OpenMaTrEx chunk aligner

OpenMaTrEx is a marker-driven example-based machine translation system (Dandapat et al., 2010)[8] that provides, among other capabilities, chunk alignment. The webservice developed performs chunk alignment by using several tools included within OpenMaTrEx, sequentially:

- Marker-based chunking (markers for all the languages of the project but Greek are available).
- Word-alignment, relying on GIZA++.
- Chunk alignment, using an algorithm based on Levenshtein edit distance,[9] and employing cognates and word probabilities as distance knowledge.

The parameters offered by the webservice are source and target language (2-character ISO codes) and source and target corpora. The output produced by the mode ebmt_alignments_on_disk_with_id, which carries out chunk alignment adding to the output sentence identifiers, has been modified to provide not only the identifier of the sentence but also the identifiers of the token that delimit each chunk in each language (OpenMaTrEx outputs the textual chunks instead of their numeric identifiers).

2.2.4 Anymalign

Anymalign is a multilingual sub-sentential aligner, which can align any number of languages simultaneously (Lardilleux and Lepage, 2010).[10] However, the webservice created is limited to a language pair, following the common interface guidelines used also for the other aligners. The parameters offered are source and target corpora and source and target languages (2-character ISO codes).

3 Travelling Object

This section deals with the conversion of the different input/output formats of the aligners from and to the Travelling Object (TO) format (defined in Deliverable 3.1, section 6.1).

7 http://code.google.com/p/berkeleyaligner/
8 http://openmatrex.org/
9 http://en.wikipedia.org/wiki/Levenshtein_distance
10 http://users.info.unicaen.fr/~alardill/anymalign/

aligner2to is a webservice that can convert the output of any[11] of the aligners that have been integrated into PANACEA to the TO. It calls the FormatConverter tool, originally developed by ILSP (see section 4 of Deliverable 4.2), which has been extended with additional processors in order to handle the formats of the aforementioned aligners.

sentalg_tok_to2word_alg is a webservice that takes three inputs in TO format (source and target corpus, both sentence-split and tokenised, and sentence alignment) and outputs two files (the subset of sentences in the source and target corpus that have 1-to-1 sentence alignments) in the plain sentence-split and tokenised format that the sub-sentential alignment webservices take as input.

sentsplit_tok2to is a webservice that was required by sentalg_tok_to2word_alg; it converts sentence-split and tokenised files to the TO format.

4 Workflows

This section presents several workflows of webservices developed in Taverna (Hull et al., 2006),[12] showing possible usages of aligners in the platform, the interactions that arise, etc.

The first workflow, depicted in Figure 1, presents a pipeline that performs sentence alignment on bilingual text crawled from the web. It is made up of two sequential workflows:

- bicrawler_to_text processes the output of the Focused Bilingual Crawler (see section 3.2 of Deliverable 4.2 for more details) for a pair of languages *A* and *B* producing three outputs: plain text versions of the source and target corpora acquired and the bilingual header in XCES format.[13]
- sentence_alignment takes as input the output from bicrawler_to_text and performs sentence alignment using Hunalign. In order to do so, the text of each language is preprocessed with a sentence splitter and a tokeniser. Finally, the workflow outputs three objects whose format is compliant with the TO: the sentence alignment and the source and target corpora sentence-split and tokenised.

11 The only exception is Anymalign, whose output is not straightforward convertible into the TO. This aligner has been integrated anyway, as it is considered useful for deriving bilingual dictionaries (work-package 5.2).

12 http://www.taverna.org.uk/

13 http://www.xces.org/schema/#align

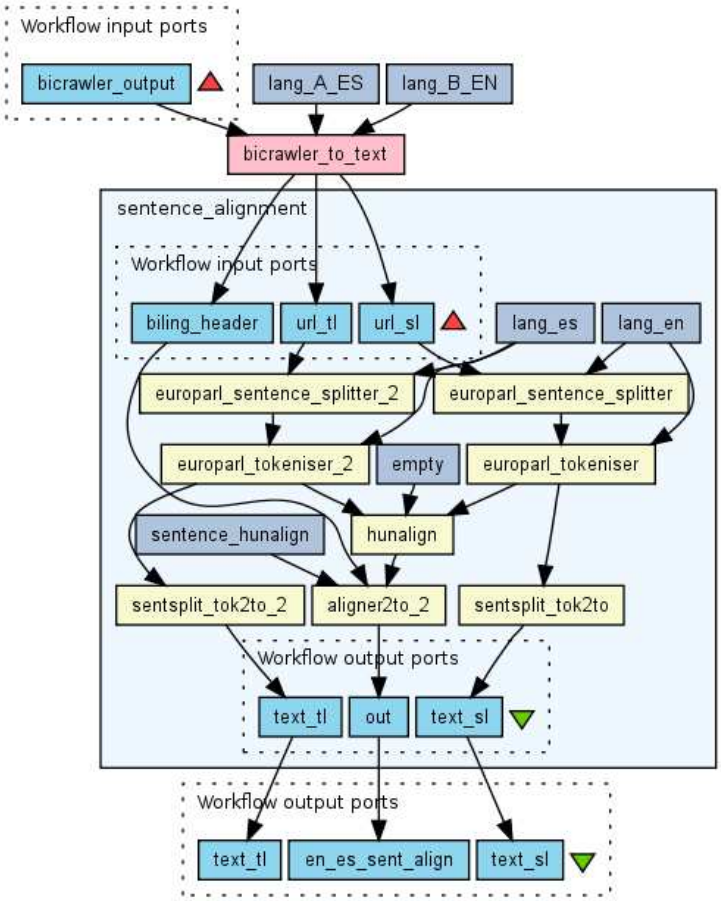

Figure 1. Sentence alignment using Hunalign

Figure 2 and Figure 3 show modified versions of the sentence_alignment workflow to perform sentence alignment with the other two tools integrated into the platform, BSA and GMA, respectively.

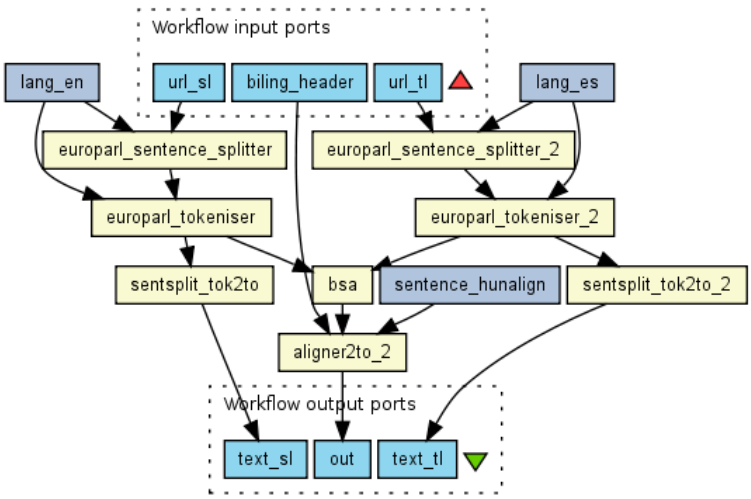

Figure 2. Sentence alignment using BSA

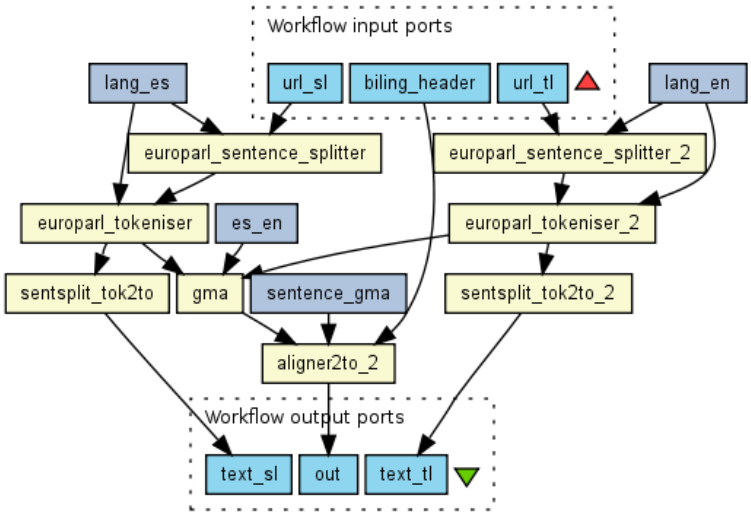

Figure 3. Sentence alignment using GMA

Figure 4 shows a workflow that performs word alignment using GIZA++.

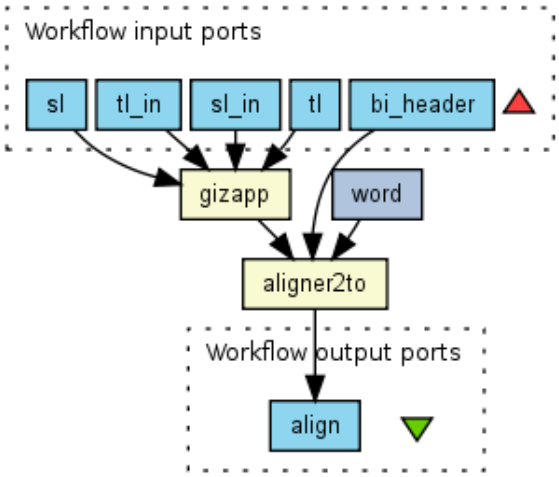

Figure 4. Word alignment using GIZA++

The workflow in Figure 5 carries out word alignment using BerkeleyAligner.

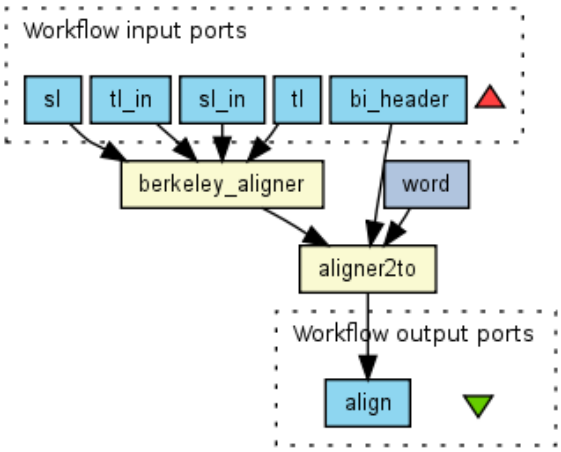

Figure 5. Word alignment using BerkeleyAligner

Figure 6 performs chunk alignment using OpenMaTrEx.

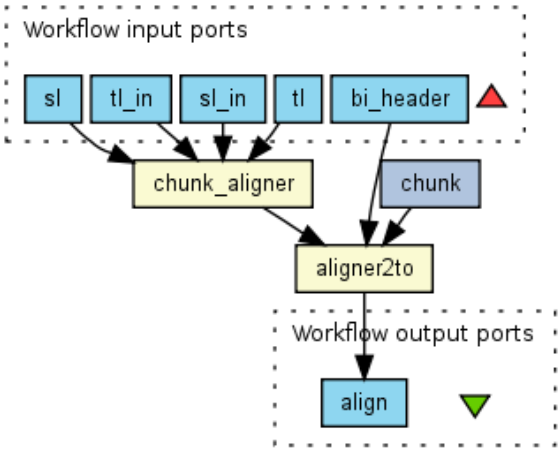

Figure 6. Chunk alignment with OpenMaTrEx

Figure 7 performs sub-sentential alignment using Anymalign.

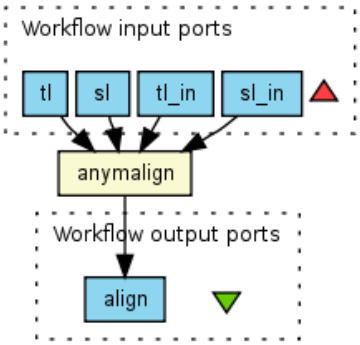

Figure 7. Sub-sentential alignment using Anymalign

Finally, Figure 8 presents a combined pipeline that performs both sentence and word alignment. It consists of a pipeline made up of four nested workflows that execute sequentially: bicrawler_to_text, sentential alignment, conversion from sentence alignment TO-compliant output to plain-text word alignment input and word alignment. The webservices used for alignment are hunalign for sentential alignment and GIZA++ for word alignment. However, any of the aligners that have been integrated into the platform could be easily used to replace these.

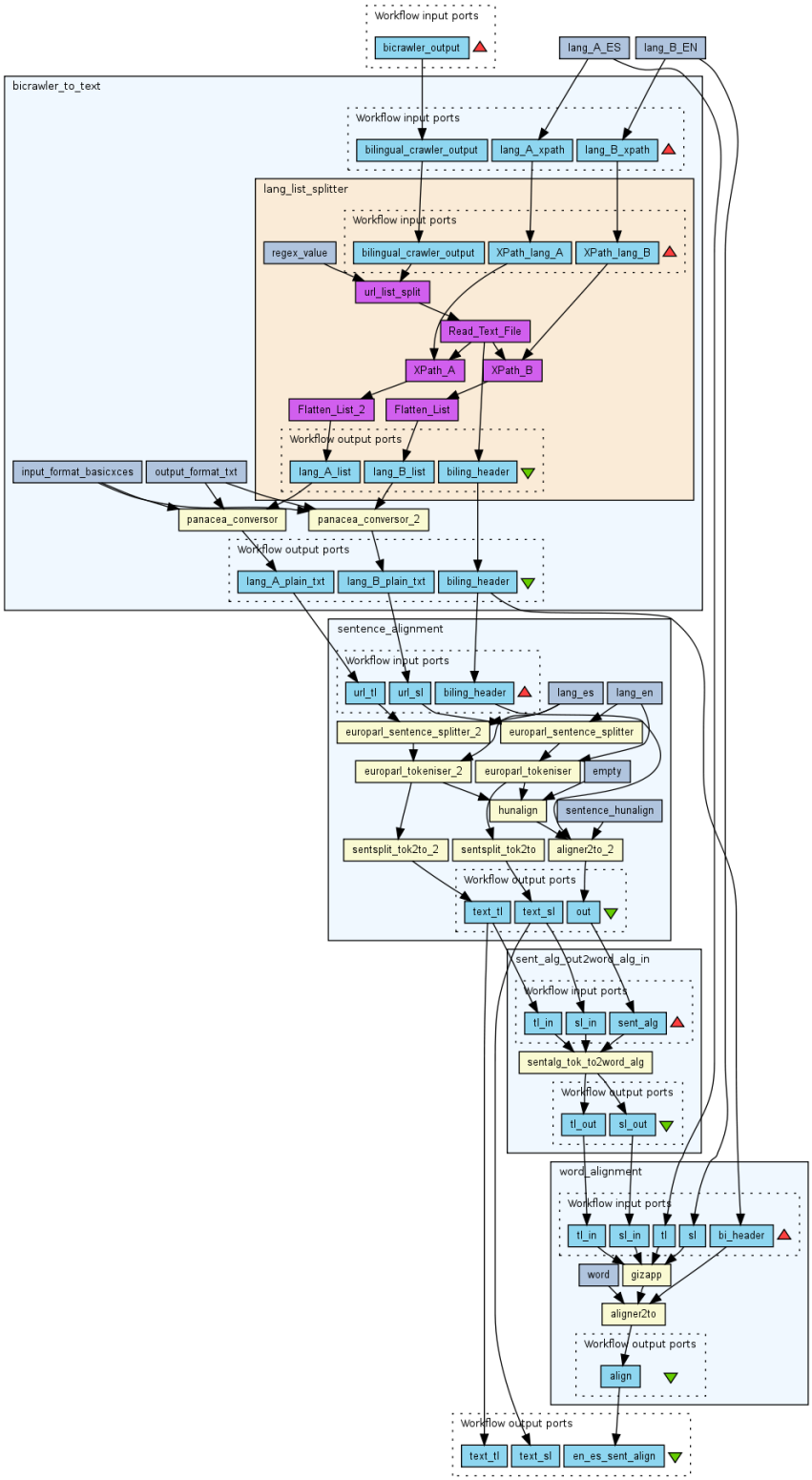

Figure 8. Pipeline that performs sentence and word alignment

5 Software

The software and data developed can be classified in two categories: webservices and workflows. IPR issues, if any, will be handled within work-package 2.

The software pipeline for each webservice reflects this schema:

ACD file → sh script → tool binary

I.e., an ACD file[14] is created to define the input/output ports of the soaplab2 webservice. This file links to a sh or perl script, which, regardless of the webservice, is in charge of three main tasks:

- Parameter handling. The parameters offered by the corresponding webservice are checked; if any of them is missing or if any different parameter is used, the execution is aborted.
- Security. Execution is automatically aborted if any command fails or if any variable is not set.
- Logging. A folder is created for each run and holds the input and output files and any log produced by the tool itself.

Finally the tool that the webservice wraps is called from the sh script.

The workflows, detailed in section 4, have been developed using Taverna 2.2. Apart from the source files, PNG exports are provided too.

All the software developed is released under the GPL version 3.[15] An archive file containing all the relevant source code is attached to this report.

6 Conclusions and Future Work

The present document has described the integration of aligners, previously discussed in Deliverable 5.1, into the Panacea platform. Webservices have been developed for each of these aligners. Each webservice acts as a wrapper over an aligner, and offers a subset of its original functionality (the aim being to provide easy-to-use tools for final users). We have discussed which functionalities to keep and which to discard according to the objectives of the platform.

Sample workflows for each of the aligners have been presented. These give a glimpse of the potential of the platform for the final user as well. They also show how the aligners interact with the other webservices of the platform.

The result of this work is the main building block for the forthcoming work on parallel aligned texts (Deliverable 5.3), as the aligners integrated in the platform will be used for this task.

14 http://soaplab.sourceforge.net/soaplab2/MetadataGuide.html
15 http://www.gnu.org/licenses/gpl-3.0.html

Bibliography

Argyle A., Shen L., Stenchikova S., and Melamed D. I. (2004.) Geometric Mapping and Alignment (GMA) tool, available at http://nlp.cs.nyu.edu/GMA/

Dandapat S., Forcada, M. L., Groves D., Penkale, S., Tinsley J., Way, A. (2010). OpenMaTrEx: A Free/Open-Source Marker-Driven Example-Based Machine Translation System. In Proceedings of IceTAL - 7th International Conference on Natural Language Processing, Reykjavík.

DeNero, J., Klein D. (2007). Tailoring Word Alignments to Syntactic Machine Translation. In proceedings of the 45th Annual Meeting of the Association for Computational Linguistics.

Haghighi, A., Blitzer, J., DeNero, J., Klein, D. (2009). Better Word Alignments with Supervised ITG Models. In Proceedings of the Joint conference of the 47th annual meeting of the Association for Computational Linguistics and 4th International Joint conference on natural language processing of the AFNLP.

D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. Pocock, P. Li, and T. Oinn. (2006). Taverna: a tool for building and running workflows of services. Nucleic Acids Research, vol. 34, iss. Web Server issue, pp. 729-732.

Adrien Lardilleux and Yves Lepage. Sampling-based multilingual alignment. International Conference on Recent Advances in Natural Language Processing (RANLP 2009), Borovets, Bulgaria, September 2009.

Liang, P., Taskar, B., Klein, D. (2006). Alignment by Agreement. In Proceedings of the Conference on Human Language Technology and Annual Meeting of the North American Chapter of the Association of Computational Linguistics.

Moore, R.C. (2002). Fast and Accurate Sentence Alignment of Bilingual Corpora, Springer-Verlag.

Och, F.J., Ney, H. (2003). "A Systematic Comparison of Various Statistical Alignment Models", Computational Linguistics, volume 29, number 1, pp. 19-51.

Varga, D., Németh, L., Halácsy, P., Kornai, A., Trón, V., Nagy, V. (2005). Parallel corpora for medium density languages. In Proceedings of RANLP 2005, pages 590-596.