SWIFT Aligner, A Multifunctional Tool for Parallel Corpora: Visualization, Word Alignment, and (Morpho)-Syntactic Cross-Language Transfer

Timur Gilmanov, Olga Scrivner, Sandra Kübler

Indiana University
Bloomington, IN, USA
{timugilm,obscrivn,skuebler}@indiana.edu

Abstract

It is well known that word aligned parallel corpora are valuable linguistic resources. Since many factors affect automatic alignment quality, manual post-editing may be required in some applications. While there are several state-of-the-art word-aligners, such as GIZA++ and Berkeley, there is no simple visual tool that would enable correcting and editing aligned corpora of different formats. We have developed SWIFT Aligner, a free, portable software that allows for visual representation and editing of aligned corpora from several most commonly used formats: TALP, GIZA, and NAACL. In addition, our tool has incorporated part-of-speech and syntactic dependency transfer from an annotated source language into an unannotated target language, by means of word-alignment.

Keywords: word alignment, parallel corpus, cross-language transfer

1. Introduction and Goals

In recent years, parallel word alignment has become a stable in machine translation, and word alignment methods such as GIZA++ (Och and Ney, 2000) have achieved significant advancement. Several factors have contributed to this achievement, namely the availability of large amounts of parallel data as well as state-of-the-art statistical algorithms in modeling and evaluation (Knight and Marcu, 2004). However, there are research areas other than machine translation, which benefit from or even depend upon word alignment. These include bilingual lexicography, multilingual word sense disambiguation, corpus-based translation studies, and cross-language transfer, among others. While machine translation takes word alignment as a given, for all the other applications, an accurate alignment on the word level is essential, and users are more willing to perform manual post-correction. Since many factors, such as genre, closeness of translation, or the distance between languages affect automatic alignment quality (Tufis, 2007), manual post-editing may be required. At present, not many tools exist to correct automatic alignment. Overall, two types of tools for correcting alignment have been introduced: 1) those that use only visual representation and 2) those that combine visual representation with editing. There are several tools that allow users to visualize word alignment pairs without making any modifications. For example, *Cairo* (Smith and Jahr, 2000), designed as an evaluation tool for the *Egypt* translation system, displays each sentence pair (source language and its translation) with lines connecting aligned words. In contrast, VisualLIHLA, part of the lexical aligner LIHLA, shows the results of alignments by highlighting aligned words (Caseli et al., 2008). Other tools offer a visualization in combination with manual annotation, for example, ILink (Merkel et al., 2003a), Yawat (Germann, 2008), COWAL (Tufis, 2006), or the UMIACS word alignment interface (Hwa and Madnani, 2004), However, all of them are restricted to a specific format of alignment. Note that those tools, being part of the larger machine translation systems, do not function as individual editors.

In this paper, we present SWIFT Aligner (short for: Speedy Word-alignment Interactive Functional Tool), a new software tool that not only allows for visual representation and editing of bitext language corpora, but also offers a number of

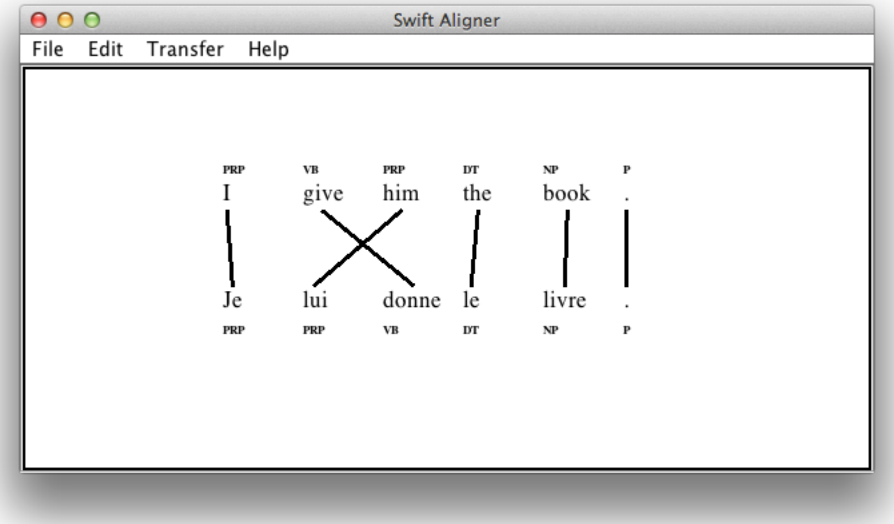

Figure 1: An example of an aligned sentence.

additional useful features:

1. **Flexibility between Alignment Formats**: Our tool is flexible in that it is not restricted to one specific format. Currently, SWIFT Aligner can import the most commonly used formats: TALP, used by the *Berkeley* aligner (Liang et al., 2006); GIZA, used by the *Giza toolkit* (Och and Ney, 2000); and NAACL, used by LIHLA (Caseli et al., 2005). In addition, SWIFT Aligner allows exporting corrected alignment into these formats as well. This flexibility to import and export various formats is a step towards bridging the gap between several machine translation tools, as it provides easier access to large data sets for linguistically oriented researchers, who are, otherwise, limited by specific format restrictions. Finally, an XML format, which is supported by a vast majority of software tools, is introduced for the purposes of internal representation for parallel alignment corpora. Consequently, this format is available for import/export purposes in SWIFT aligner.

2. **Interactivity**: Our tool is simple, intuitive to use, and interactive. While there are several visualization techniques for parallel alignment, namely word matrices, different coloring schemes for word pairs (see e.g. (Germann, 2008)), or enumerating links between each word pair, we chose the most common and simple technique: drawing lines between pairs of corresponding words. An example of the alignment visualization is shown in Figure 1. The visual interface allows a user to correct the alignment by dragging these lines to the correct source and target word pairs.

3. **Multifunctionalilty**: We have incorporated automatic part-of-speech (POS) and syntactic dependency annotation via cross-language transfer (see Figure 2 for an example on the POS level). The transfer is based on the word alignment. Thus, the user can import annotations for one (source) language, transfer these annotations to the target language, and work on correcting the transferred annotations manually. Post-transfer manual annotation, depending on the user preferences, can be performed inside SWIFT Aligner. Our tool provides support for manual creation of

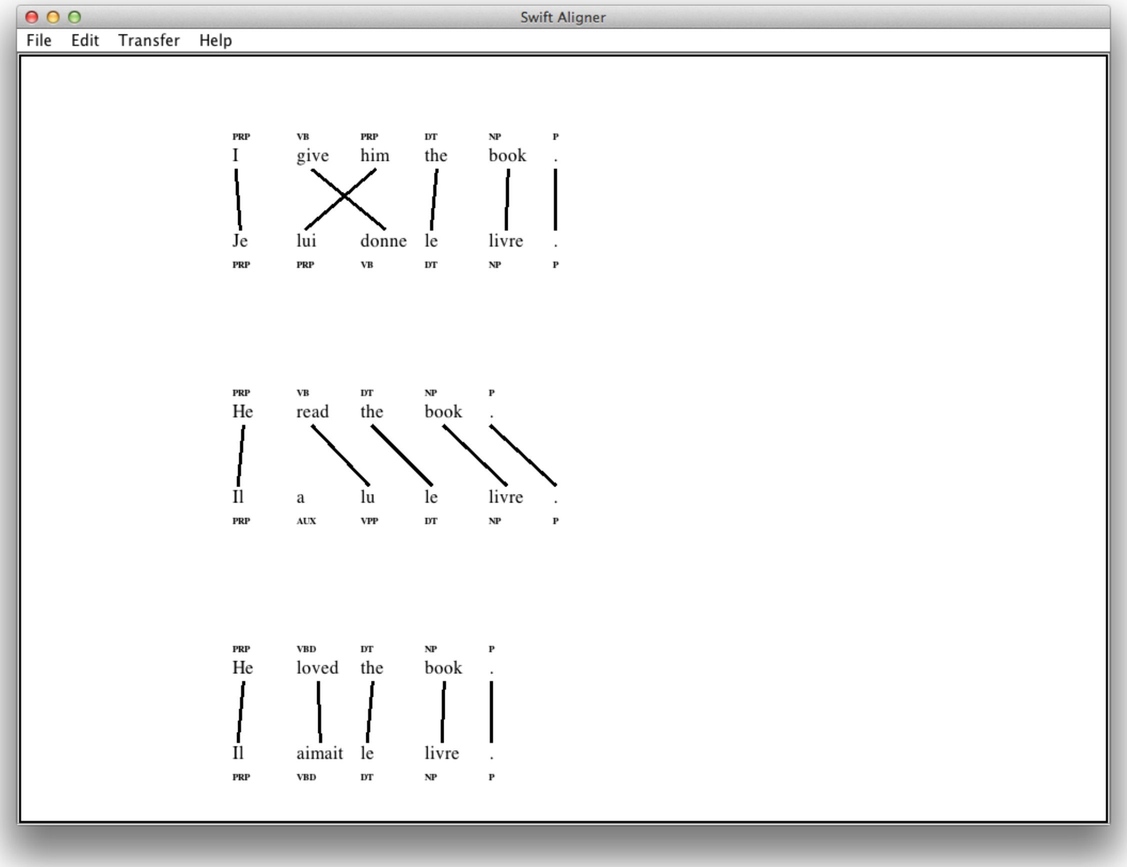

Figure 2: SWIFT aligner's main GUI. Alignment between words in 3 different sentences is presented. POS labels are displayed for each word.

POS and syntactic dependency annotation, and the user can use the same GUI in order to perform the required annotation. However, if a user prefers to perform these corrections outside of SWIFT Aligner, it is possible to export the aligned and annotated text.

4. **Platform Independence**: SWIFT Aligner is implemented in Java (Java SE 6), thus providing a multi-platform functionality without a difficult installation procedure. The code is written in a modular way. This allows for easy understanding of the code as well as for extensibility. For example, introducing new data formats for exporting/importing purposes, is possible. For such extensions, pre-defined code templates exist and can easily be extended. An elementary familiarity with Java is required, however, in order to add such new functionality. Additionally, the tool supports UTF-8 representations, which allows for support of a wide range of languages, including right-to-left languages.

2. Related Work

There is a considerable body of work on alignment for machine translation. Our work is more closely related to approaches that focus on visualizing and post-correcting alignment. Thus, we focus our discussion of related work on work that is relevant in this regard.

The combination of automatic alignment and manual post-editing has been introduced in recent alignment tools. One of the first such tools is *Interactive Linker* (Ahrenberg et al., 2002). This tool, and its later version *I*Link*, (Merkel et al., 2003b) are built for an interactive incremental alignment process, allowing the human annotator to adjust links between word pairs during automatic alignment. The *Combined Word Aligner*, COWAL (Tufis, 2006), is a combination of two aligners and a graphical user interface for intermediary and final alignment correction. There are also several tools that focus on the manual editing of alignments provided by the tool, for example, *Yawat* (Germann, 2008), COWAL, or the UMIACS word alignment interface (Hwa and Madnani, 2004). Note that none of the above listed tools allow imports from other machine translation alignment systems. Each of the mentioned tools for the visualization and editing maintains its own internal format. Since there are several commonly used formats, these tools are difficult to use in a more generalized setting. In the following, we will briefly describe these formats alongside with the XML format that we use for internal representation of parallel corpora in SWIFT aligner. All of the discussed formats represent the alignment shown in Figure 1.

The first format that can be used with SWIFT Aligner is the TALP format. An example is shown in Figure 3. The alignment is represented in three separate files, one each for the source text, the target text, and the alignment information. The separate files are represented as individual lines in the example. The third line in Figure 3 encodes the alignment between words, the first number referring to the source sentence and the second number to the target sentence. Thus, the alignment 2-3 specifies that the second source word `give` is aligned with the third target word `donne`.

The NAACL format, presented in Figure 4, is similar to TALP in that it stores the alignment in three separate files, which we represent by separate lines respectively. In this format, the alignment is represented by referring to the source word rather than its index. To assign a unique id to each sentence in these files, an XML-like tagging for sentence numbers is used.

The GIZA format stores the alignment information in one file, with each sentence represented by two lines, the source (second line) and target sentence (first line), with the alignment encoding in parentheses. Thus, the word `give` in Figure 5 is aligned with the third word in the target sentence in the first line.

Finally, the XML format stores all the alignment information in one file, as shown in Figure 6. First, the sentence id is denoted, which is followed by a source and target encodings.

3. Cross-Language Transfer

The idea of cross-language transfer by means of word-alignment is not new. Over the past years, several studies have demonstrated the usability of parallel corpora for automatic (morpho-)syntactic transfer from a source language into a target language. One of the advantages of this method is the creation of NLP resources for less-common languages. The other positive outcome lies in the enhancement of machine translation system since many parallel corpora aligners achieve better accuracy when syntactically annotated texts are available. The feasibility of transfer methods has been tested in several realms of machine translation: for example, parallel bilingual parsing, part-of-speech transfer and part-of-speech tagger induction, noun phrase bracketer induction, syntactic transfer, and word sense disambiguation, among others (Yarowsky and Ngai, 2001; Lin, 1998; Wu, 1997; Tufis, 2006). In the area of part-of-speech transfer and sense annotation transfer, the transfer algorithm is traditionally based on a direct label projection from a source into a target language. For example, Yarowsky and Ngai (2001) describe an experiment for morphosyntactic transfer from English into French. The results show that morph-syntactic annotation can be effectively transferred using a small core tagset. In the area of syntactic transfer, it has been shown that syntactic dependencies are more suitable than syntactic constituents for cross-language syntactic transfer (Lin, 1998). Many models of syntactic dependency transfer follow a principle of direct correspondence assumption (DCA), which specifies that syntactic relations between nodes of the source language hold for the corresponding

```
I give him the book .
Je lui donne le livre .
1-1 2-3 3-2 4-4 5-5 6-6
```

Figure 3: TALP format example

```
<s snum=1>I give him the book .</s>
<s snum=1>Je lui donne le livre .</s>
<s snum=1>I:1 give:3 him:2 the:4 book:5 .:6</s>
```

Figure 4: NAACL format example

```
Je lui donne le livre .
NULL ({ }) I ({ 1 }) give ({ 3 }) him ({ 2 }) the ({ 4 }) ...
... book ({ 5 }) . ({ 6 })
```

Figure 5: GIZA format example

```
<sentence id="1">
<Source>
<word align="1" form="I" id="1"/>
<word align="3" form="gave" id="2"/>
<word align="2" form="him" id="3"/>
<word align="4" form="the" id="4"/>
<word align="5" form="book" id="5"/>
<word align="6" form="." id="6"/>
</Source>
<Target>
<word form="Je" id="1"/>
<word form="lui" id="2"/>
<word form="donne" id="3"/>
<word form="le" id="4"/>
<word form="livre" id="5"/>
<word form="." id="6"/>
</Target>
</sentence>
```

Figure 6: XML format example

aligned nodes of the target language (Hwa et al., 2005). Several experiments have evaluated the DCA model for different languages. For example, Hwa et al. (2005) performed several experiments on languages with different word order, namely English, Spanish, and Chinese. While the direct projection from English yielded low unlabeled dependency F-scores (37% for Chinese and 38% for Spanish), the errors mostly occurred in cases where the target language required more projections than the source language (English). For instance, Chinese aspectual markers are not realized as a separate projection in English; therefore, they are left unlabeled during the transfer. The application of simple language-specific transformation, however, increased the accuracy to 68% for Chinese and 72% for Spanish transfer.

In SWIFT aligner, both POS and syntactic depen-

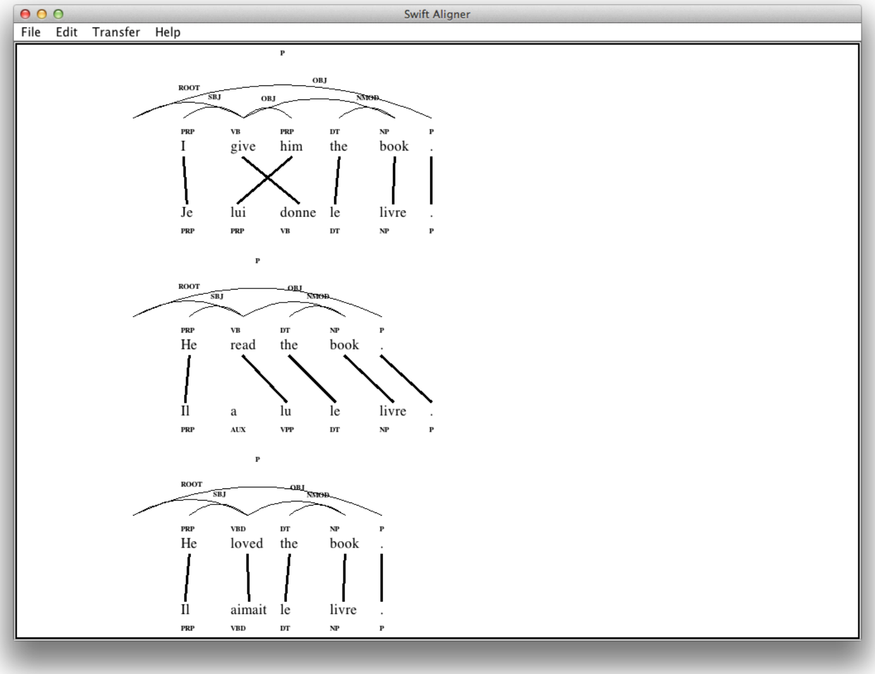

Figure 7: An example of aligned sentences with POS tags and source language dependencies displayed.

dencies transfer procedures are available. The user can import POS annotations, dependency annotations, or both for the source language. Then, the system transfers the annotation to the target language. We follow a simple procedure in which the POS and dependencies are, essentially, assigned to the aligned structures in target language. Thus we are applying the direct correspondence assumption by Hwa et al. (2005). A situation where no alignment exists is presently resolved by assigning a tag ”NA” to the target structure. This can then be manually post-corrected by the user (see next section). Note that currently, our focus is not on improving current cross-transfer strategies but rather to provide a user-friendly tool that allows for an easy inspection and correction of the transfer results.

4. Correcting Linguistic Annotation

In order to further enhance SWIFT Aligner for linguistically oriented end users, we include the option of editing the POS tags and syntactic dependencies via the GUI, see Figure 7. The combination of annotation post-editing and alignment enhances the efficiency of manual correction as the user is able to compare target and source languages visually. To correct POS tags, the user can click on the the POS tags field and then correct the POS tags by typing in the correct version. Since in a cross-language transfer situation, it is possible that the user may want to introduce new POS tags for the target language, we do not check for consistency. Instead, the user has the possibility of accessing a list of all POS tags for each language. To correct the dependencies, the user can drag the dependency arc in the same way as for the alignment. To correct dependency labels, the strategy is the same as for POS tags.
When the desired correction state is reached, the new results can be saved to the preferred output format.

5. Conclusions and Future Work

We have introduced a new tool for parallel corpora visualization, editing, and (morpho-)syntactic cross-language transfer. This tool is intuitive and easy to use. It can be beneficial to researchers working with parallel corpora, as well as the broader linguistic community in cases where quick annotations of target languages are required. For the future, we are planning to integrate state-of-the-art cross-language transfer algorithms for POS tagging and dependency parsing. This will include the integration of a POS tagger and a dependency parser, which can be retrained on the target language annotations.

6. References

Lars Ahrenberg, Mikael Andersson, and Magnus Merkel. 2002. A system for incremental and interactive word linking. In *Third International Conference on Language Resources and Evaluation (LREC)*, pages 485–490, Las Palmas, Gran Canaria.

Helena M. Caseli, Maria G. V. Nunes, and Mikel L. Forcada. 2005. LIHLA: Shared task system description. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 111–114, Ann Arbor, MI.

Helena Caseli, Felipe T. Gomes, Thiago A.S. Pardo, and Maria das Graças V. Nunes. 2008. VisualLIHLA: The visual online tool for lexical alignment. In *Companion Proceedings of the XIV Brazilian Symposium on Multimedia and the Web*, pages 378–380, Vila Velha, Brazil.

Ulrich Germann. 2008. Yawat: Yet another word alignment tool. In *Proceedings of the ACL-08*, pages 20–23, Columbus, OH.

Rebecca Hwa and Nitin Madnani, 2004. *The UMIACS word alignment interface*. University of Maryland Institute for Advanced Computer Studies.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11(3):311–325.

Kevin Knight and Daniel Marcu. 2004. Machine translation in the year 2004. In *Proceedings of ICASSP*, Montréal, Canada.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, HLT-NAACL '06, pages 104–111, New York, NY.

Dekang Lin. 1998. Dependency-based evaluation of minipar. In *Proceedings of the LREC Workshop on the Evaluation of Parsing Systems*, Granada, Spain.

Magnus Merkel, Lars Ahrenberg, and Michael Pettersted. 2003a. Interactive word alignment for language engineering. In *Proceedings of the Tenth Conference of the European Chapter of the ACL*, pages 49–52, Budapest, Hungary.

Magnus Merkel, Michael Pettersted, and Lars Ahrenberg. 2003b. Interactive word alignment for corpus linguistics. In *In Proceedings of Corpus Linguistics 2003*, Lancaster, UK.

Franz Josef Och and Hermann Ney. 2000. A comparison of alignment models for statistical machine translation. In *Proceedings of the 18th Conference on Computational Linguistics*, pages 1086–1090, Saarbrücken, Germany.

Noah A. Smith and Michael E. Jahr. 2000. Cairo: An alignment visualization tool. In *Second International Conference on Linguistic Resources and Evaluation (LREC-2000).*

Dan Tufis. 2006. From word alignment to word senses, via multilingual wordnets. In *Computer Science Journal of Moldova*, volume 14, pages 3–33.

Dan Tufis. 2007. Exploiting aligned parallel corpora in multilingual studies and applications. In *Proceedings of the 1st International Conference on Intercultural Collaboration*, pages 103–117, Kyoto, Japan.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. In *Computational Linguistics*, volume 23, pages 377–404.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Proceedings of the 2nd Meeting of NAACL*, pages 1–8, Pittsburgh, PA.