PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 94 SEPTEMBER 2010 97–106

A Toolkit for Visualizing the Coherence of Tree-based Reordering with Word-Alignments

Gideon Maillette de Buy Wenniger, Maxim Khalilov, Khalil Sima'an

Institute for Logic, Language and Computation, University of Amsterdam

Abstract

Tree-based reordering constitutes an important motivation for the increasing interest in syntax-driven machine translation. It has often been argued that tree-based reordering might provide a more effective approach for bridging the word-order differences between source and target sentences. One major approach (known as Inversion Transduction Grammar) allows permuting the order of the subtrees dominated by the children of any node in the tree. In practice, it has often been observed that the word-alignments usually cohere only to a certain degree with this kind of tree-based reordering, i.e., there are cases of word-alignments that cannot be fully explained with tree-based reordering *when the tree is fixed a priori*.

This paper describes a toolkit for visualizing alignment graphs that consist of a word-alignment together with a source or target tree. More importantly, the toolkit provides a facility for visualizing the coherence of word-alignment with tree-based reordering, highlighting nodes and word-alignments that are incompatible with one another. The tool allows visualizing the tree-based reordered source/target string as well as the reordered tree.

1. Introduction

Word-alignment (the mapping from source language words to target language words) is the starting point for most translation systems in the Statistical Machine Translation (SMT) (e.g. (Och and Ney, 2004; Koehn et al., 2003; Mariño et al., 2006)). Such translation relationships among the words can be m-to-n in the most general case, were m source words can produce n target words.

General tools have been created for the visualization of basic word alignment (Smith and Jahr, 2000; Germann, 2008) as well as for the manual annotation of

Corresponding author: gmaillet@science.uva.nl
Cite as: Gideon Maillette de Buy Wenniger, Maxim Khalilov, Khalil Sima'an. A Toolkit for Visualizing the Coherence of Tree-based Reordering with Word-Alignments. The Prague Bulletin of Mathematical Linguistics No. 94, 2010, pp. 97–106. doi: 10.2478/v10108-010-0024-4.

sentence pairs. The recent trend in the SMT research community towards including syntactic information into SMT systems makes the simultaneous visualization of alignment and parse trees increasingly more important. This combination of source or target tree with alignment links is also known as an *alignment graph* (Galley et al., 2004). One other toolkit became recently available which focuses on the alignment of parallel treebanks, the *Stockholm Tree Aligner* (STA) (Volk et al., 2007).

Beyond the mere visualization of alignment graphs, there is currently an increasing need for visualizing the coherence of tree-based reordering with word-alignment for the purposes of understanding word-order divergence phenomena between pairs of languages. Our tree and alignment visualization toolkit (TAVT) is aimed exactly at this functionality. TAVT allows visualizing alignment graphs as well as the extent to which word-alignments and the constituency parse tree are compatible with one another. More concretely, TAVT visualizes how well reordering under the ITG constraints (Wu, 1997), while restricting the tree to be the source constituency parse (Yamamoto et al., 2008), succeeds to arrive at monotonic word-alignments, i.e. resolves all crossing word-alignments. Our toolkit allows visualizing the source/target permutation that can be obtained under the assumption of ITG-based tree-reordering as well as the incompatible nodes and word-alignments. This is especially useful for the researchers working on tree-to-string or tree-to-tree translation, as this is exactly the data their systems are built upon.

2. Tree and Alignment Coherence Visualizer

In this section we describe our visualization toolkit TAVT. The first function of our toolkit is the visualization of basic word alignments without trees. However, the main and distinguishing function of TAVT is the simultaneous visualization of trees and alignments (*alignment graphs*). Finally, our toolkit enables the automatic reordering of the source tree to optimally match the target string word order under ITG constraints, and allows visualizing the resulting permutation as well as the nodes and alignments incompatible with one another.
TAVT is implemented using parts of the code extracted from the package ConstTree-Viewer – a constituency structures viewer, available on:
`http://staff.science.uva.nl/~fsangati/`.

TAVT itself can be downloaded from:
`http://code.google.com/p/tree-alignment-visualizer/`.

2.1. Basic Alignment Visualization

Visualizing the basic word alignments is done by displaying the two sentences one beneath the other, with the aligned words being connected by colored lines. For ease of reading in case of cluttered and crossing alignment lines, different colors are used

for the alignments of the different words (see Figure 1 for an example of a human made m to n alignment).

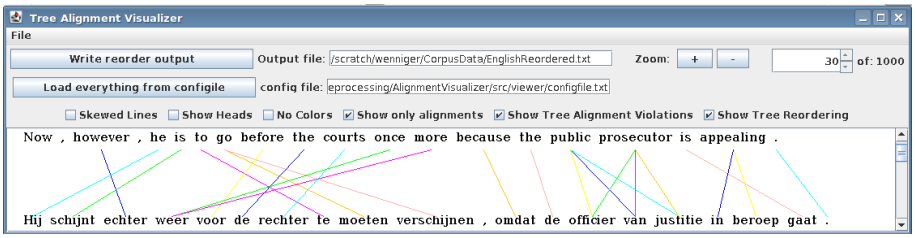

Figure 1: Visualization of the basic word alignments between a source and target sentence.

In our further examples we have restricted ourselves to the *intersection* of the GIZA++ alignments in two directions, which is a common method to improve precision at the price of recall. Furthermore we use IBM model 1 lexical weights to keep only the strongest alignment of a source word, in case it is aligned to multiple target words. These choices are arbitrary and other choices could have been made just as well. The resulting alignments are only 1-to-1 or 1-to-0, and as a result of the intersection they are also incomplete. While different operations such as union allows construction of general *m*-to-*n* alignment, the modeling restriction of word-based alignment made by GIZA++ introduces inherent artifacts that direct tree-based alignment could overcome (Pauls et al., 2010).

2.2. Visualizing the Alignment Graph

The visualization of the *alignment graph* is similar to the visualization of the basic word alignments. Rather than displaying the source words, this visualization shows the source tree ending in the leaf nodes containing source words. The source words are again connected by lines to their target-side counterparts.

Basic phrase-based machine translation systems work with phrase pairs that are consistent with the word alignment: the words in a legal phrase pair are contiguous strings consisting of words aligned to each other and not to words outside. As a refinement, syntactic phrases are phrases that are covered by a single subtree in the constituency parse tree (Koehn et al., 2003). From the point of view of syntactic SMT, it is very interesting to see what subtrees of the (source) parse tree are *alignment cohesive* (henceforth *cohesive*), i.e. correspond to the source side of a syntactic phrase pair, and what constituents root a set of children that fail to form a contiguous phrase on the target side. The distinguishing feature of the TAVT is that it gives insight into the *alignment cohesiveness* of subtrees of the parse tree, and *un-cohesiveness* which occurs when alignment spans for subtrees overlap. The overlaps imply that tree-constrained

reordering fails in the sense that ITG-based tree transductions are not sufficient to achieve the ultimate reordering goal: a reordered source tree that has no crossing alignments and whose lexical order matches the order of the target sentence.

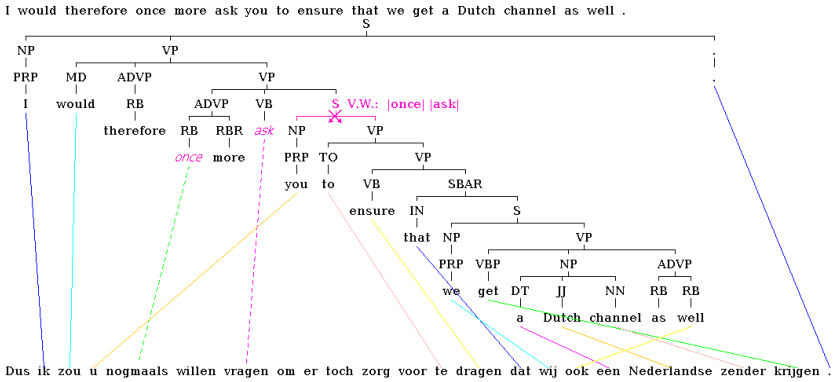

Figure 2: Visualization of the alignment graph.

Tree Alignment Violations and *Cohesive* Nodes. We define overlapping alignment spans, as follows:

Let $A(n) \rightarrow \{1, \cdots, m\}^*$ be the alignment mapping function that maps a source leaf node to a set of zero or more of the m possible target alignment positions.

A certain subtree rooted at node n in the alignment graph *spans* a range $AlignmentSpan(n)$ of target positions defined by the minimum and maximum alignment position over its descending leaf source words:

Definition 2.1 (Alignment Span)
$AlignmentSpan(n) :=$

$$[a_{n_{min}}, a_{n_{max}}] = \left[\min_{x \in \text{LeafNodes}(n)} \left(\min_{a_{x'} \in A(x)} a_{x'} \right), \max_{y \in \text{LeafNodes}(n)} \left(\max_{a_{y'} \in A(y)} a_{y'} \right) \right]$$

Note that every source leaf node is in principle allowed to map to multiple target word positions, so we have to perform a double minimization/maximization to get the minimum and maximum over this set of sets of alignment positions.

Any source leaf node n' that is not descending from n but aligns to a word in the same range $[a_{n_{min}}, a_{n_{max}}]$ is said to "violate" the *alignment span* of n.

Definition 2.2 (Alignment Violation)
$violates(n', n) \coloneqq terminal(n') \wedge n' \notin \text{descendants}(n) \wedge$
$(AlignmentSpan(n) = [a_{n_{min}}, a_{n_{max}}]) \wedge (a_{n_{min}} \leq A(n') \leq a_{n_{max}})$

A node n is said to be *cohesive* if it has no alignment violation, in other words that node alone aligns to the contiguous target side of its associated phrase pair.

Alignment violations are indicated by displaying every "violated" subtree in the alignment graph in purple, and showing a list of violating words (V.W.) behind the node label. A symbol consisting of two crossing arrows, just below the root of every subtree that is violated further emphasizes the alignment violations. The violating word itself is accented by an italic purple font, and its alignment is drawn as a striped line to further emphasize its overlap with the alignment range of the other subtree (see Figure 2).

2.3. Optimal tree-constrained source reordering

Displaying how the source tree can be optimally reordered to match the target word order is the goal of the third component of our visualization toolkit. In the tree reordering, we assume that the original parse tree structure must be preserved, the same assumption as made in (Khalilov and Sima'an, 2010). The only allowed operation is the permutation of the child nodes under a parent node (Yamamoto et al., 2008). Given this limited reordering freedom and the alignment spans of different nodes, the tree nodes can be reordered to get a modified tree that better matches the target word order. To do so, every non-terminal node in the tree is visited and every pair of child nodes $c1$ and $c2$ is compared. $c1$ moves before $c2$ if and only if the alignment span of $c1$ precedes the alignment span of $c2$, denoted as $AlignmentSpan(c1) < AlignmentSpan(c2)$ and defined as:

Definition 2.3 (Alignment Span Precedence)
$AlignmentSpan(c1) = [a1_{\min}, a1_{\max}] < AlignmentSpan(c2) = [a2_{\min}, a2_{\max}]$
$\coloneqq (a1_{\min} < a2_{\min}) \wedge (a1_{\max} < a2_{\min})$

Two alignment spans can only be compared if they do not overlap and the one span strictly begins and ends before the other. Note that this is automatically the case if at least one of the two source nodes associated with these alignment spans is *cohesive*. And so the circle closes. *Cohesive* nodes are important since they imply a reordering is possible that will put the source phrases covered by these nodes at the right position, matching the target word position of that phrase (but not necessarily recursively the right order *within* the phrase). *Alignment Violations* are similarly important, since they imply *un-cohesiveness* and thus show were the tree-based reordering scheme fails, be it for alignment errors or simply linguistic complexities.

The example in Figure 3 illustrates the reordering visualization. It shows an alignment graph with multiple crossing alignments due to the fact that the phrase

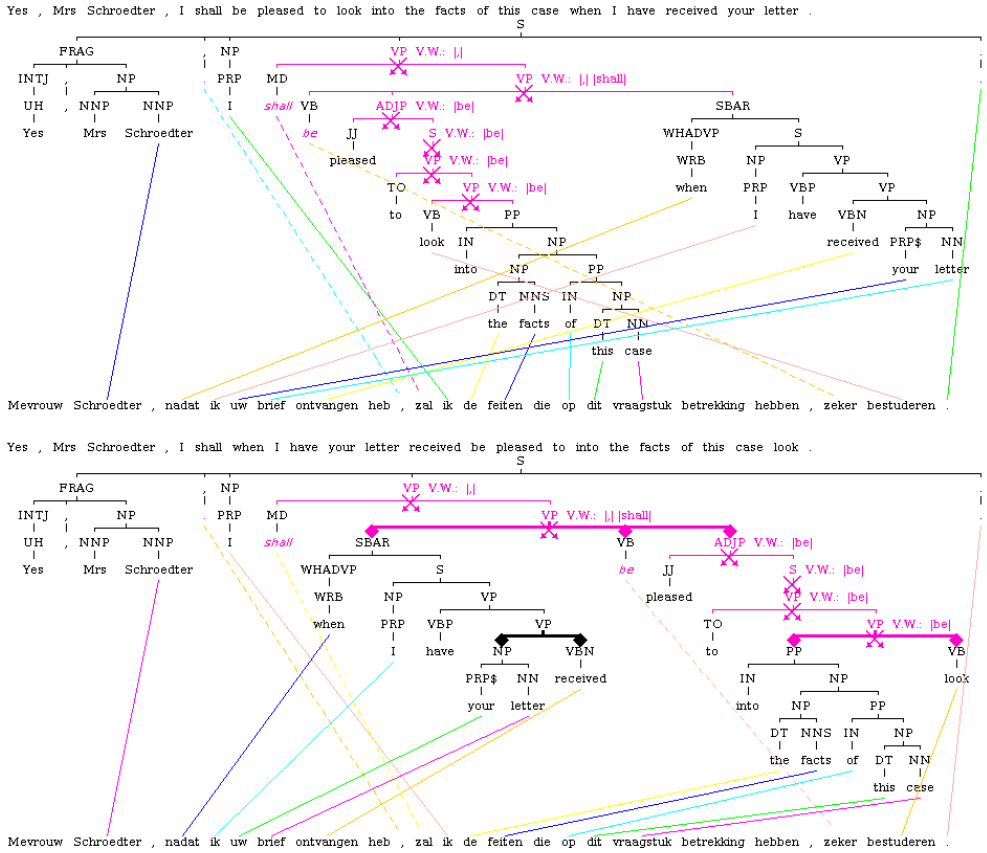

Figure 3: Visualization of the tree-constrained reordering.

"when I have received your letter" → *"nadat ik uw brief ontvangen heb"* moves from the end of the English source sentence almost completely to the beginning in the Dutch translation. Furthermore there are three words : *","* , *"shall"* and *"be"* that cause alignment violations with other subtrees in the alignment graph. In the reordered tree, a thick horizontal line indicates that some of the child nodes under a subtree root are reordered, while those child nodes that really moved to a new position are further emphasized by a diamond just above the node. Unaligned source words like "have" in the example do not directly constrain the word order. However,when the subtrees they belong to move to a new position, they move with them, so their position in the reordered tree is just as well unambiguous. Notice, that in the subtree that roots the phrase *"be pleased ··· received your letter"* not all the children have comparable

alignment spans since the subtree covering *"pleased to look into the facts of this case"* has an alignment span that overlaps with that of its sibling node that covers *"be"*. In contrast, the alignment span of the other sibling covering *"when I have received your letter"* strictly precedes the span of both other siblings, and is thus moved to the front. Therefore, the word order becomes much more like the target word order by performing the reordering procedure, even though overlapping alignment ranges prevent all crossing alignments from being resolvable.

3. Usability

Visualization is generally an effective way for representing and (re-)organizing multi-source information. TAVT is a toolkit that intends to manage word alignment and syntactic information and help users process translation content more efficiently. At the design level, we tried to make our toolkit as intuitive and easy to use as possible. The entire toolkit is written in Java and requires no installation of external libraries. Different aspects of TAVT usability are considered below:

Alignment visualization. At the most basic level TAVT is convenient to browse easily through the different alignments and corresponding parse trees in the data set. This gives a lot of insight in the data in relatively short time and consequently helps in designing effective translation systems.

Alignment and tree visualization. The visualization of trees in addition to the basic alignments helps in different ways. Subtrees are expected to be aligned as word blocks most of the time, if this does not happen it is a clear indication of discrepancy between word alignment and syntactic bilingual segmentation, or alternatively an alignment or parse error.

In this concern, our work has a fair amount of overlap with the STA tool, presented in (Volk et al., 2007). However the focus and functionality of the TAVT toolkit and STA tool differ significantly. While STA visualizes parallel treebanks with alignments, TAVT visualizes alignment graphs which assume only one tree is available. For translation this is often a more realistic assumption, since for many languages no reliable parsers exist. Our goals in developing this application are centered on getting insight into the parallel corpus, dealing with alignment problems, and marking where and how the word reordering takes place and what kind of tree transformations could support it. Therefore we provide functionality for the visualization of tree reordering and order conflicts resulting from subtrees with overlapping alignment spans. Another difference is that STA requires an available parallel treebank, in which case it is very useful, however in practice this implies such a treebank must be (manually) build. In contrast, our toolkit works with automatically extracted

information: GIZA++ alignment and parse trees produced by any constituency parser.

Tree-constrained word reordering visualization. One more important field of TAVT application is the word reordering task at the pre-translation step (Collins et al., 2005; Costa-jussà and Fonollosa, 2006; Xia and McCord, 2004). Here, the word reordering problem is attacked by introducing the pre-processing step into the SMT system, in which the input is rearranged in order to make the source sentence word order resemble that of the target language. Many of these reordering systems exploit syntactic representations of source and target texts and that is where our visualization toolkit can be an asset. The visualization of reordering by means of child node permutations gives a precise idea about how far one can get towards a corpus free of crossing alignments with this transformation to the target word order.

4. Conclusions

TVTA is an open-source visualization toolkit targeted especially to researchers, developers and students working in the field of SMT. Our toolkit goes beyond what other visualization tools offer: by the incorporation of trees in the visualization TVTA gives a lot of meaningful information that other alignment visualization tools do not provide. The extra information is expected to be useful in the research towards better translation systems, and in testing whether certain hypotheses about the translation patterns of a certain language pair actually hold in the data.

5. Future Work

The TVTA visualization framework presents many opportunities for future work. In this section, we describe some of the paths we wish to investigate in the future.

Heuristics in subtree reordering. How should one decide which of two child nodes with overlapping alignment spans should go first in the parse tree? Currently we are preserving their order as it is, since the overlap in alignment spans causes these spans to be *incomparable*. However, this might not always be the best way to deal with it. We have some ideas how we might define order preferences for such incomparable alignment spans. One idea would be to take the alignment weights into account in combination with the target positions of the aligned words in the two spans, and then define a heuristic reordering preference based on these. This is especially important if we want to go beyond mere visualization and learn reordering rules such as it is done in (Khalilov and Sima'an, 2010).

Tree modification. Another idea for an extension would be to incorporate other tree transduction operations, such as insertion and deletion of nodes. Transformation by

a minimum number of such operations would produce a transformed source tree $\tau_{s'}$ that roots the source order in a new order such that the alignments are all non-crossing, while staying as close as possible to the original tree. Visualization of such operations would give a good idea about what combination of operations allows what level of alignment disentanglement. Then with this insight, a more optimal trade-off between level of coverage and computational complexity could be made.

Alignment refinement. Continuing in the same direction, yet another idea is to explore the change of alignments through local re-alignment operations or a whole extra re-alignment phase in the translation process such as described by (Wang et al., 2010). Rather than blindly following the alignments and transforming our trees to match them, we should take into consideration the fact that certain alignments are wrong and the tree can help us to find these. Indeed, recent research in alignment and machine translation builds on the insight that alignments and tree transductions should be optimized simultaneously instead of being factored into two independent steps (Burkett et al., 2010; Pauls et al., 2010). Extensions to the visualization toolbox may indeed be helpful to get insight into the effect of (automatic) re-alignment operations, possibly in combination with simultaneous tree reordering, and how this helps to improve the translation process.

Acknowledgement

This project was supported in part by the Project "Machine Translation When Exact Pattern Match Fails" as part of "free competition for the exact sciences" funded by the Dutch organization for scientific research (NWO) under project number 612066929. The authors wish to thank Federico Sangati who was so kind to make his tree visualization software available on top of which our toolkit has been built.

Bibliography

Burkett, D., J. Blitzer, , and D. Klein. Joint parsing and alignment with weakly synchronized grammars. In *Proc. of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2010.

Collins, M., P. Koehn, and I. Kučerová. Clause restructuring for statistical machine translation. In *Proc. of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, pages 531–540, 2005.

Costa-jussà, M. R. and J. A. R. Fonollosa. Statistical machine reordering. In *Proc. of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 70–76, 2006.

Galley, M., M. Hopkins, K. Knight, and D. Marcu. What's in a translation rule? In *Proc. of the Human Language Technology Conference and the North American Association for Computational Linguistics (HLT-NAACL)*, pages 273–280, 2004.

Germann, U. Yawat: yet another word alignment tool. In *Proc. of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies (ACL-HLT)*, pages 20–23, 2008.

Khalilov, M. and K. Sima'an. A discriminative syntactic model for source permutation via tree transduction. In *Proc. of The Fourth Workshop on Syntax and Structure in Statistical Translation (SSST-4) at the 23rd International Conference on Computational Linguistics (COLING)*, pages – to appear, 2010.

Koehn, P., F.J. Och, and D. Marcu. Statistical phrase-based translation. In *Proc. of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL)*, 2003.

Mariño, J. B., R. E. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. R. Fonollosa, and M. R.Costa-jussà. N-gram based machine translation. *Computational Linguistics*, 32(4):527–549, 2006.

Och, F. and H. Ney. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30:417–449, 2004.

Pauls, A., D. Klein, D. Chiang, and K. Knight. Unsupervised syntactic alignment with inversion transduction grammars. In *Proc. of 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2010)*, 2010.

Smith, Noah A. and Michael E. Jahr. Cairo: An alignment visualization tool. In *Proc. of the 2nd Conference on Language Resources and Evaluation (LREC)*, page 549551, 2000.

Volk, M., J. Lundborg, and M. Mettler. Alignment tools for parallel treebanks. In *In Proc. of The Linguistic Annotation Workshop at the Association for Computational Linguistics (LAW-ACL)*, 2007.

Wang, W., J. May, K. Knight, and D. Marcu. Re-structuring, re-labeling and re-aligning for syntax-based machine translation. *Computational Linguistics*, 36:247–277, 2010.

Wu, D. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23:377–404, 1997.

Xia, F. and M. McCord. Improving a statistical MT system with automatically learned rewrite patterns. In *Proc. of the 20th international conference on Computational Linguistics (COLING)*, pages 508–514, 2004.

Yamamoto, H., H. Okuma, and E. Sumita. Imposing constraints from the source tree on itg constraints for smt. In *Proc. of the Second Workshop on Syntax and Structure in Statistical Translation (SSST '08)*, page 19, 2008.

Address for correspondence:
Gideon Maillette de Buy Wenniger
gmaillet@science.uva.nl
Institute for Logic, Language and Computation
University of Amsterdam
Science Park 904
1098 XH Amsterdam, Netherlands