Robust Language Pair-Independent Sub-Tree Alignment

John Tinsley, Ventsislav Zhechev, Mary Hearne and Andy Way

National Centre for Language Technology, Dublin City University, Dublin, Ireland
{jtinsley, vzhechev, mhearne, away}@computing.dcu.ie

Abstract

Data-driven approaches to machine translation (MT) achieve state-of-the-art results. Many syntax-aware approaches, such as Example-Based MT and Data-Oriented Translation, make use of tree pairs aligned at sub-sentential level. Obtaining sub-sentential alignments manually is time-consuming and error-prone, and requires expert knowledge of both source and target languages. We propose a novel, language pair-independent algorithm which automatically induces alignments between phrase-structure trees. We evaluate the alignments themselves against a manually aligned gold standard, and perform an extrinsic evaluation by using the aligned data to train and test a DOT system. Our results show that translation accuracy is comparable to that of the same translation system trained on manually aligned data, and coverage improves.

1. Introduction

The majority of approaches to data-driven Machine Translation (MT) focus on string-to-string models, despite the fact that tree-to-tree models achieve promising results (Hearne & Way, 2006; Nesson et al., 2006). Some tree-to-tree models, such as Data-Oriented Translation (DOT) (Poutsma, 2003; Hearne & Way, 2003, 2006), require source and target tree pairs that are aligned at sub-sentential level. In most previous experiments with DOT systems the training tree pairs were aligned manually. However, such a task is time-consuming and error-prone, and requires considerable expertise in both the source and target languages, and so there is an obvious need to induce the alignments automatically.

A considerable amount of research has been carried out on the subject of sub-sentential alignment between structured representations of sentence pairs. However, many of the solutions presented share one or both of the following characteristics: (i) the alignment process is tightly coupled with the intended application, to the extent that it is difficult to see how to generalise the alignment methodology so that the output could be used for other applications; (ii) the alignment strategy edits the source and/or target linguistic representations such that the original linguistic structures cannot be retrieved.

We present a novel, language pair-independent and task-independent algorithm whose output may be useful in many applications. The algorithm induces alignments between paired linguistic structures from which the constituent surface word order can be determined. It handles complex, non-isomorphic structures in a fast and consistent manner, and the resulting output can be ported to many other translation tasks such as Phrase-Based Statistical MT, Example-Based MT, DOT and translation template extraction.

We describe experiments where we apply our algorithm to context-free phrase-structure tree pairs. We evaluate the alignments themselves against a manually aligned gold-standard, and also perform an extrinsic evaluation by using the aligned data to train and test a DOT system. Our results show that translation accuracy is comparable to that of the same translation system trained on manually aligned data from English to French, and coverage improves significantly.

The remainder of this paper is organised as follows: Section 2 details related work and in Section 3 we present our novel alignment algorithm. Section 4 describes our experiments including the MT system used, and finally in Sections 5 and 6 we conclude and discuss avenues for further research.

2. Related Work

Previous approaches to automatic sub-sentential alignment can be loosely grouped according to whether they focus on aligning dependency structures or phrase-structure trees. Many approaches do not view alignment as an independent task, but rather as a means to achieving another goal such as solving parse ambiguities or acquiring translation templates. Some such approaches view factors like non-isomorphism as obstacles, and alter the trees as part of the alignment process.

Other related work in the area of alignment in general views the use of tree structures as a negative aspect which may result in the loss of generalisation ability (Wellington et al., 2006). However, we chose to align pre-determined tree structures without editing them; our motivation is that the structural and translational divergences that exist between source and target structures should be captured during the alignments process rather than smoothed away in order to allow for higher recall, cf. (Hearne et al., 2007). We do not view the parse trees as constraints, but rather as accurate syntactic representations of the text which can help to guide the alignment process.

2.1. Dependency Structures

We are particularly interested in aligning phrase-structure trees, but solutions which have been applied to the alignment of dependency structures are also relevant.

Ding et al. (2003) present a strategy for inducing word alignments over dependency structures. However, dependency analyses contain only lexically headed phrases, and we want to capture links between non-lexically headed phrases not described in dependency representations.

Matsumoto et al. (1993) induce alignments in dependency structures, but with the intention of using the align-

ments to resolve parse ambiguities. Their algorithm only aligns simple sentences: alignment of complex sentences is done by first segmenting the sentence into smaller chunks and then aligning those chunks, and the original tree structures are not retrievable.

Eisner (2003) develops a tree-mapping algorithm for use on dependency structures which he claims is adaptable for use on phrase-structure trees. However, the alignment process is heavily linked to the translation strategy of which it forms part. We prefer alignment to be a separate offline process which can then be applied to numerous different tasks.

2.2. Phrase-structure Trees

Groves et al. (2004) present a rule-based aligner which builds upon automatically induced word alignments. While their algorithm is in theory language pair-independent, in later experiments it performed poorly when evaluated on language pairs other than those used in development.

Gildea (2003) proposes a method for aligning non-isomorphic phrase-structure trees using a stochastic tree-substitution grammar (STSG). This approach involves the altering of the tree structure in order to impose isomorphism, which impacts on its portability to other domains.

Lu et al. (2001) describe a stochastic inversion transduction grammar, based on (Wu, 1995), which uses a monolingual grammar to parse the source sentence and builds a target language parse based on this, while simultaneously inducing alignments. These alignments are then extracted and converted into translation templates. Imposition of source language structure onto the target language is not always desirable. Nevertheless, on the evidence presented here, tree-to-string alignment models warrant further investigation.

Wang et al. (2002) develop an interesting method for structural alignment which they call "bilingual chunking". Given a pair of phrase-structure trees, they perform word alignment on the surface forms and then extract chunks from both trees simultaneously. The chunking is guided by the tree structure and constraints which ensure word alignments do not cross chunks. The chunks are then POS-tagged using an HMM tagger. Again, the original tree structures are lost during the alignment process.

While the methods outlined above all achieve competitive results, those presented by Lu et al. (2001) and Wang et al. (2002) are most closely aligned with our objectives.

3. Our Sub-Tree Alignment Algorithm

The novel algorithm we present here is designed to discover an optimal set of alignments between the tree pairs in a bilingual treebank while adhering to the following principles:

(i) independence with respect to language pair and constituent labelling schema;
(ii) preservation of the given tree structures;
(iii) minimal external resources required;
(iv) word-level alignments not fixed *a priori*.

The algorithm makes use of a single external resource, namely target-to-source and source-to-target word translation probabilities generated by running an automatic word aligner over the sentence pairs encoded in the bilingual treebank. The algorithm does not, however, fix *a priori* on a single word-alignment between the source and target terminals of each sentence pair. Rather, word-level alignment decisions can be influenced by links made higher up in the tree pair. The alignment algorithm does not edit or transform the source and target trees in any way; significant structural and translational divergences are to be expected and the aligned tree pair should encode these divergences (Hearne et al., 2007). Finally, the algorithm accesses no language-specific information beyond the (automatically induced) word-alignment probabilities and does not make use of the node labels in the tree pairs.

3.1. Alignment Well-Formedness Criteria

Links are induced between tree pairs such that they meet the following well-formedness criteria:

(i) a node can only be linked once;
(ii) descendants of a source linked node may only link to descendants of its target linked counterpart;
(iii) ancestors of a source linked node may only link to ancestors of its target linked counterpart.

These criteria are akin to the "crossing constraints" described in (Wu, 1997) which forbid alignments between constituents that cross each other. Our criteria differ from those of Wu because we impose them on a pair of fully monolingually parsed trees, thus our criteria are more strict. The constraints in (Wu, 1997), on the other hand, are imposed inherently during the bilingual parsing and alignment process.

In what follows, a hypothesised alignment is ill-formed with respect to the existing alignments if it violates any of these criteria.

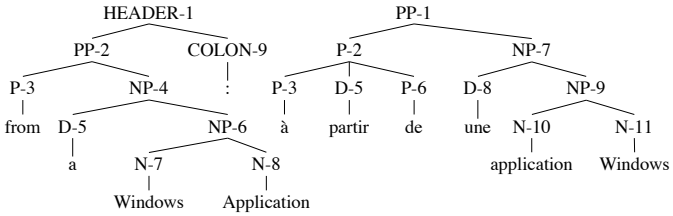
3.2. Algorithm

In this section we describe how our algorithm scores and selects links. In some instances, we present alternative methods by which a decision can be taken, and at the end of the section we summarise the corresponding set of possible aligner configurations.

3.2.1. Selecting Links

For a given tree pair $\langle S, T \rangle$, the alignment process is initialised by proposing all links $\langle s, t \rangle$ between nodes in S and T as hypotheses and assigning scores $\gamma(\langle s, t \rangle)$ to them. All zero-scored hypotheses are blocked before the algorithm proceeds. The selection procedure then iteratively fixes on the highest-scoring link, blocking all hypotheses that contradict this link and the link itself, until no non-blocked hypotheses remain. These initialisation and selection procedures are given in **Algorithm 1** *basic*.

Figure 1 illustrates the **Algorithm 1** *basic* procedure. The constituents in the source and target tree pair are numbered. The numbers down the left margin of the grid correspond to the source constituents while the numbers across the top correspond to the target constituents, and each cell in the grid corresponds to a scored hypothesis. Within each cell, circles denote selected links and brackets denote blocked links. The number inside a given cell indi-



	1	2	3	5	6	7	8	9	10	11
1	(1)	0	0	0	0	0	0	0	0	0
2	(1)	0	0	0	0	0	0	0	0	0
3	0	(3)	0	0	0	0	0	0	0	0
4	0	0	0	0	0	(6)	0	0	0	0
5	0	0	0	0	(2)	0	(2)	0	0	0
6	0	0	0	0	0	(2)	0	(5)	(4)	0
7	0	0	0	0	(3)	0	0	0	0	(7)
8	0	0	0	0	0	0	0	0	(4)	0
9	0	0	0	0	(3)	0	(2)	0	0	(5)

Figure 1: Illustration of how **Algorithm 1** *basic* induces links for the tree-pair on the left.

cates the iteration during which its link/block decision was made, with zeroes indicating hypotheses with score zero. For example, hypothesis ⟨1, 1⟩ was linked during iteration 1, and hypothesis ⟨2, 1⟩ was blocked, hypothesis ⟨5, 8⟩ was linked during iteration 2 and hypotheses ⟨5, 6⟩, ⟨6, 7⟩ and ⟨9, 8⟩ were blocked, and so on. There were 7 iterations in total, and the last iteration linked the remaining non-zero hypothesis ⟨7, 11⟩.

Algorithm 1 *basic*

Initialisation

```
for each source non-terminal s do
  for each target non-terminal t do
    generate scored hypothesis γ(⟨s, t⟩)
  end for
end for
block all zero-scored hypotheses
```

Selection *underspecified*

```
while non-blocked hypotheses remain do
  link and block the highest-scoring hypothesis
  block all contradicting hypotheses
end while
```

Hypotheses with equal scores The selection procedure given in **Algorithm 1** *basic* is incomplete as it does not specify how to proceed if two or more hypotheses share the same highest score. We propose two alternative solutions to this problem. Firstly, we can simply skip over tied hypotheses until we find the highest-scoring hypothesis with no competitors of the same score, as given by **Algorithm 2 Selection** *skip1*.

Algorithm 2 Selection *skip1*

```
while at least one non-blocked hypothesis with
    no tied competitors remains do
  while the highest-scoring hypothesis has
      tied competitors do
    skip
  end while
  link and block the highest-scoring
      non-skipped hypothesis
  block all contradicting hypotheses
  re-enable all non-blocked skipped hypotheses
end while
```

The skipped hypotheses will, of course, still be available during the next iteration, assuming that they have not been ruled out by the newly-selected link. If all but one of the tied hypotheses have been ruled out, the remaining one will be selected on the next iteration. If all remaining non-zero-scored hypothesis have tied competitors then no further links can be induced.

A second alternative is to skip over tied hypotheses until we find the highest-scoring hypothesis ⟨*s*, *t*⟩ with no competitors of the same score and *where neither* s *nor* t *has been skipped*, as given in **Algorithm 3 Selection** *skip2*.

Algorithm 3 Selection *skip2*

```
while at least one non-blocked hypothesis with
    no tied competitors remains do
  if the highest-scoring hypothesis has
      tied competitors then
    mark the constituents of all competitors
        as skipped
  end if
  while the highest-scoring hypothesis has
      a skipped constituent do
    skip
  end while
  link and block highest-scoring
      non-skipped hypothesis
  block all contradicting hypotheses
  re-enable all non-blocked skipped hypotheses
end while
```

This alternative is proposed in order to avoid the situation in which a low-scoring hypothesis for a given constituent is selected in the same iteration as higher-scoring hypotheses for the same constituent were skipped, thereby preventing one of the competing higher-scoring hypotheses from being selected and resulting in an undesired link. The issue is illustrated in Figure 2, as follows. The best-scoring hypotheses, of which there are several, involve source constituent D-21 and include the correct hypothesis ⟨D-21, D-16⟩. The *skip1* solution simply selects the best non-tied hypothesis, ⟨D-21, D-4⟩, which is clearly incorrect. The *skip2* solution, however, skips over all hypotheses involving skipped constituent D-21 and selects ⟨D-16, D-4⟩ as the best hypothesis. On the next iteration, all hypotheses for source constituent D-21 are again skipped, and hypothesis ⟨PP-18, PP-13⟩ is selected. This selection blocks all but one hypothesis involving source constituent D-21, the correct hypothesis ⟨D-21, D-16⟩, and so this link is selected on the following iteration.

Delaying span-1 alignments It is frequently the case that the highest-scoring hypotheses are at the word level, i.e. have span 1 on the source and/or target sides. However, selecting links between frequently occurring lexical

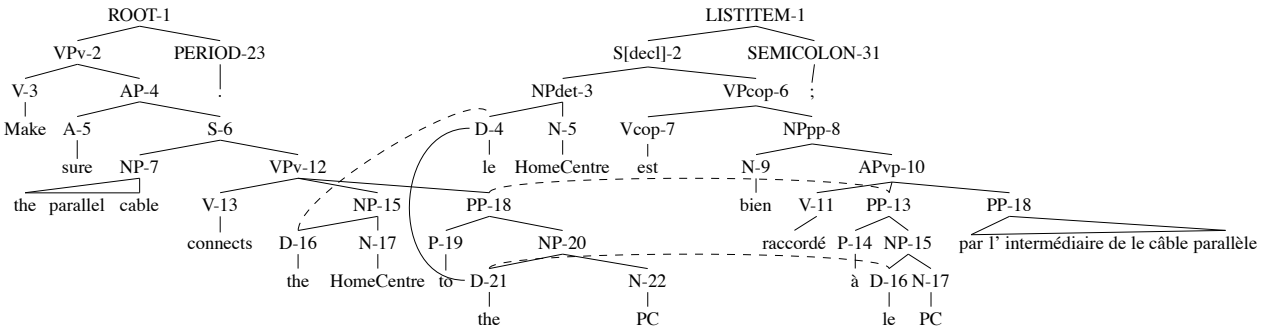


Figure 2: This example illustrates the differing effects of the **Selection** *skip1* and **Selection** *skip2* strategies: with *skip1* the solid link is induced whereas with *skip2* the dashed links are induced.

items at an early stage is intuitively unappealing. Consider, for instance, the situation where source terminal *x* most likely translates to target terminal *y* but there is more than one occurrence of both *x* and *y* in a single sentence pair. It may be better to postpone the decision as to which instance of *x* corresponds to which instance of *y* until links higher up in the tree pair have been established, as given in **Algorithm 4 Selection** *span1* (where span-1 hypotheses have span 1 on the source and/or target sides and non-span-1 refers to all other hypotheses).

Algorithm 4 Selection *span1*

```
while non-blocked non-lexical hypotheses remain
    do
  link and block the highest-scoring hypothesis
  block all contradicting hypotheses
  if no non-blocked non-lexical hypotheses
      remain then
    while non-blocked lexical hypotheses remain
        do
      link and block the highest-scoring
          hypothesis
      block all contradicting hypotheses
    end while
  end if
end while
```

The effects of the **Selection** *span1* strategy are illustrated by the example given in Figure 3: without *span1*, node D-8 is immediately linked to D-13 rather than D-4 and D-17 to D-4 rather than D-13. Not only are these alignments incorrect, but their presence means that the remaining desirable hypotheses are no longer well-formed. However, the correct alignments are induced by first allowing NP-7 to link to NP-3 and NP-16 to NP-12.

3.2.2. Computing Hypothesis Scores

Inserting a link between two nodes in a tree pair indicates that (i) the substrings dominated by those nodes are translationally equivalent and (ii) all meaning carried by the remainder of the source sentence is encapsulated in the remainder of the target sentence. The scoring method we propose accounts for these indications.

Given tree pair $\langle S, T \rangle$ and hypothesis $\langle s, t \rangle$, we compute the following strings:

$$s_l = s_i \ldots s_{ix} \qquad \overline{s_l} = S_1 \ldots s_{i-1} s_{ix+1} \ldots S_m$$
$$t_l = t_j \ldots t_{jy} \qquad \overline{t_l} = T_1 \ldots t_{j-1} t_{jy+1} \ldots T_n$$

where $s_i \ldots s_{ix}$ and $t_j \ldots t_{jy}$ denote the terminal sequences dominated by s and t respectively, and $S_1 \ldots S_m$ and $T_1 \ldots T_n$ denote the terminal sequences dominated by S and T respectively. These string computations are illustrated in Figure 4.

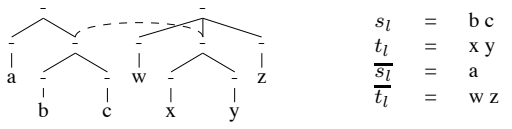


Figure 4: Values for s_l, t_l, $\overline{s_l}$ and $\overline{t_l}$ given a tree pair and a link hypothesis.

The score for the given hypothesis $\langle s, t \rangle$ is computed according to (1).

$$\gamma(\langle s, t \rangle) = \alpha(s_l|t_l) \cdot \alpha(t_l|s_l) \cdot \alpha(\overline{s_l}|\overline{t_l}) \cdot \alpha(\overline{t_l}|\overline{s_l}) \quad (1)$$

Individual string-correspondence scores $\alpha(x|y)$ are computed using word-alignment probabilities given by the

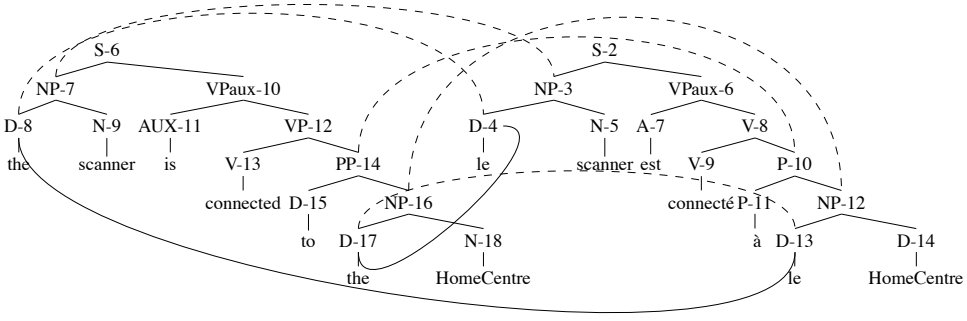


Figure 3: This example illustrates the effects of the **Selection** *span1* strategy: without *span1* the solid links are induced whereas switching on *span1* results in the dashed alignments.

Moses decoder[1] (Koehn et al., 2007). To improve the quality of the word-alignments, we induce them from a lowercased version of the parallel corpus, but our algorithm does not change the case of the data.

Two alternative scoring functions are given in (2) and (3). They differ in that *score2* divides the sum over the probabilities that x_i corresponds to each y_j by the number of words in y. The intended effect of this is again to reduce any bias in favour of aligning shorter-span constituents over constituents of longer span.

(2) **Score** *score1*
$$\alpha(x|y) = \prod_{j}^{|y|} \sum_{i}^{|x|} \mathrm{P}(x_i|y_j)$$

(3) **Score** *score2*
$$\alpha(x|y) = \prod_{i}^{|x|} \frac{\sum_{j}^{|y|} \mathrm{P}(x_i|y_j)}{|y|}$$

3.3. Aligner Configurations

When configuring the aligner, we must choose either *skip1* or *skip2* and we must choose either *score1* or *score2*. *span1* is optional, and so can be switched either on or off. The eight possible configurations are as follows:

skip1_score1	*skip1_score1_span1*
skip1_score2	*skip1_score2_span1*
skip2_score1	*skip2_score1_span1*
skip2_score2	*skip2_score2_span1*

4. Evaluation

We perform an intrinsic evaluation by comparing the links induced by the alignment algorithm against a manually-aligned gold standard. We also perform an extrinsic evaluation by using these alignments to train a DOT system and then measuring translation quality.

We evaluate the performance of our sub-tree alignment algorithm on the English-French section of the HomeCentre corpus, which contains 810 parsed, sentence-aligned translation pairs.[2] This corpus comprises a Xerox printer manual, which was translated by professional translators and sentence-aligned and annotated at Xerox PARC. As one would expect, the translations it contains are of extremely high quality.

We ran the unlinked tree pairs through the eight configurations of our alignment algorithm given in Section 3.3[3]. The manual alignments were provided by a single annotator, who is a native English speaker with proficiency in French (Hearne, 2005).

4.1. Intrinsic Evaluation

In this section, we evaluate precision and recall of induced alignments over the 810 English-French tree pairs described above, using the manually linked version as a gold standard.

Given a tree pair T, its automatically-aligned version T_A and its manually-aligned version T_M, precision and recall are computed as given in (4) and (5).

(4)
$$Precision = \frac{|T_A \cap T_M|}{|T_A|}$$

(5)
$$Recall = \frac{|T_M \cap T_A|}{|T_M|}$$

In addition to calculating the precision and recall over all links, we also calculate scores over non-lexical links only, where a non-lexical link aligns constituents which both span more than one word. We do so in order to determine how successful our algorithm is at inducing alignments beyond the word level. Table 1 gives the results of this evaluation for the different configurations of the aligner.

Configurations	*all links*		*non-lexical links*	
	Precision	Recall	Precision	Recall
skip1_score1	0.6096	0.7723	**0.8424**	0.7394
skip1_score2	0.6192	0.7869	0.8107	0.7756
skip2_score1	0.6162	0.7783	0.8394	0.7486
skip2_score2	0.6215	0.7867	0.8107	0.7756
skip1_score1_span1	0.6229	**0.8101**	0.8137	0.7998
skip1_score2_span1	0.6220	0.7963	0.8027	0.7871
skip2_score1_span1	**0.6256**	0.8100	0.8139	**0.8002**
skip2_score2_span1	0.6245	0.7962	0.8031	0.7871

Table 1: Evaluation of the automatic alignments against the manual alignments.

Looking firstly to the *all links* column, it is immediately apparent that recall is significantly higher than precision for all configurations. In fact, we have noted that all aligner variations consistently induce more links than exist in the manual version, with the average number of links per tree pair ranging between 10.3 and 11.0 for the automatic alignments versus 8.3 links per tree pair for the manual version. Regarding the differences in performance between the aligner variants, we observe that all versions which include *span1* outperform all versions which exclude it. When *span1* is excluded *score2* performs better than *score1*, but this is reversed once *span1* is introduced. No clear difference is shown between *skip1* and *skip2* – *skip2* performs marginally better than *skip1*.

Looking now to the *non-lexical links* column, we observe that the balance between precision and recall is reversed and that precision is now higher than recall in all cases. This indicates that those phrase-level alignments we induce are reasonably accurate and suggests that, conversely, the accuracy of our lexical-level alignments is relatively poor. Regarding the differences in performance between the aligner variants, we note that both highest

[1] Although our method of scoring is similar to IBM model 1, and Moses runs GIZA++ trained on IBM model 4, we found that using the Moses word-alignment probabilities yielded better results than those output directly by GIZA++.

[2] The average numbers of English and French words per sentence are 8.83 and 10.05 respectively, and the average numbers of English and French nodes per tree are 15.33 and 17.52 respectively.

[3] The aligner takes approx. 0.01 seconds per tree pair on an Apple MacBook Pro with a 2.33GHz Intel Core 2 Duo processor and 2GB of RAM; time variations over aligner configurations are insignificant.

precision and lowest recall are achieved using *skip1_score1* and *skip2_score1*. However, the best balance between precision and recall is again achieved when the *span1* option is used.

Another option for intrinsic evaluation could have been a comparison with a state-of-the-art word- or phrase-alignment system. We decided against such an evaluation, because of the inherent differences in the way such systems induce alignments compared to our algorithm. This will be further discussed in section 4.2.2. Often, for such alignments there are no corresponding constituents in the syntactic trees that could be linked by the aligner presented here. Although, as mentioned in section 2, we do not see this as a drawback, this impedes direct comparison. Our experiments show that only 83.6% of the sentence pairs in the HomeCentre corpus have phrase alignments that represent constituents in the syntactic trees and among those there are only 3.4 such alignments per sentence pair on average. Our algorithm creates links matching on average 70-80% of the phrase alignments that are constituents depending on the configuration, but we do not regard these results as an indication of the quality of the aligner. A comparison to a word-aligner would be even less indicative, because the many-to-many word-alignments such a system produces only rarely represent constituents in a phrase-structure tree.

Further evaluation carried out in (Hearne et al., 2007) discusses the performance of the aligner with respect to capturing translational divergences between the treebank languages.

4.2. Extrinsic Evaluation

In this section, we train and test a DOT system using the manually aligned data introduced above, and we evaluate the output translations to give us baseline scores. We then train the system on the automatically aligned data and repeat the same tests, such that the only difference across runs is the alignments.

4.2.1. The MT System

Data-Oriented Translation (DOT) (e.g. (Poutsma, 2003; Hearne & Way, 2006)), which is based on Data-Oriented Parsing (DOP) (e.g. (Bod et al., 2003)), combines examples, linguistic information and a statistical translation model. Tree-DOT assumes training data in the form of aligned source-target context-free phrase-structure tree pairs, such as the one given in Figure 5(a), from which it learns a generative model of translation. This model takes the form of a synchronous stochastic tree-substitution grammar (S-STSG) whereby pairs of linked generalised subtrees are extracted from the linked tree pairs contained in the training data via root and frontier operations.

- given a copy of tree pair $\langle S, T \rangle$ called $\langle S_c, T_c \rangle$, select a **linked** node pair $\langle S_N, T_N \rangle$ in $\langle S_c, T_c \rangle$ to be *root* nodes and delete all except these nodes, the subtrees they dominate and the links between them, and
- select a set of **linked** node pairs in $\langle S_c, T_c \rangle$ to be *frontier* nodes and delete the subtrees they dominate.

Thus, every fragment $\langle f_s, f_t \rangle$ is extracted such that the root nodes of f_s and f_t are linked, and every non-terminal frontier node in f_s is linked to exactly one non-terminal frontier node in f_t and vice versa. Some fragments extracted from Figure 5(a) are given in Figure 5(b).

During translation, fragments are merged in order to form a representation of the source string within which a target translation is embedded. The composition operation ($\circ$) is a leftmost substitution operation: where a fragment has more than one open substitution site, composition must take place at the leftmost site on the source subtree of the fragment. Furthermore, the synchronous target substitution must take place at the site *linked to* the leftmost open source substitution site. This ensures (i) that each derivation is unique and (ii) that each translation built adheres to the translational equivalences encoded in the example base. An example composition sequence is given in Figure 5(c).

Many different representations and translations can be generated for a given input string, and the alternatives are ranked using a probability model. In the system used for these experiments, fragment probabilities are estimated using relative frequencies and derivation probabilities computed by multiplying the probabilities of the fragments used to build them. For each input string, the n-best derivations are generated and then reduced to the m-best translations where the probability of translation t is computed by summing over the probabilities of those derivations that yield it. Where no derivation spanning the full input string can be generated, the n-best sequences of partial derivations are generated instead and the translations ranked as above. Unknown words are simply left in their

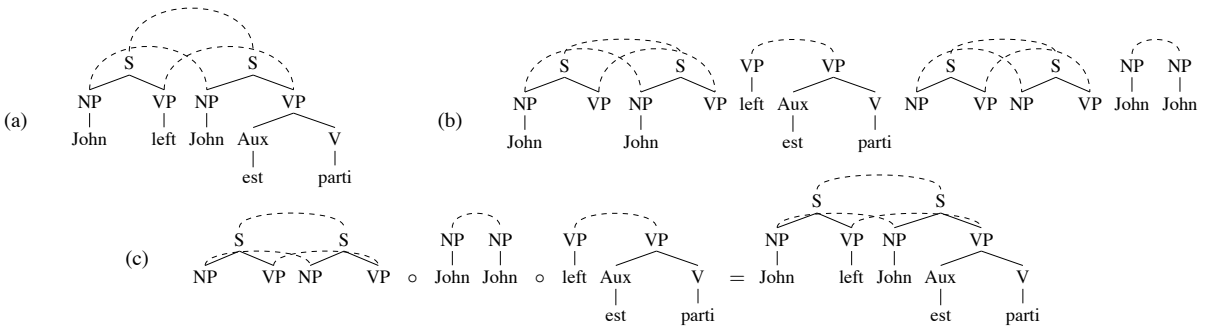


Figure 5: Data-Oriented Translation: (a) gives an example representation, (b) gives a subset of the possible fragments of (a) and (c) gives an example composition sequence yielding a bilingual representation.

source form in the target string. Thus, every input string is translated but the system output indicates which strings achieved full coverage.

4.2.2. Experiments and Results

We again used 9 versions of the HomeCentre dataset, one aligned manually and the others using the aligner configurations specified in Section 3.3. We also generated 6 training/test splits for the dataset at random such that (i) all test words also appeared in the training set, (ii) all splits have English as the source language and French as the target language and (iii) each test set contains 80 test sentences each training set contains 730 tree pairs. We then applied the 6 splits to each of the 9 versions of the dataset, trained the MT system on each training set and tested on each corresponding test set. We evaluated the translation output using three automatic evaluation metrics, BLEU (Papineni et al., 2002), NIST (Doddington, 2002) and METEOR (Banerjee & Lavie, 2005), averaging the results over the 6 splits in order to gain a single score for each of the 9 variants of the aligned dataset. We also measured coverage for each variant.

These scores are presented in Table 2. We note that most of the automatically aligned runs outperform the manual scores for the BLEU and METEOR metric and that the NIST scores for the automatic alignments are very competitive. Furthermore, all the automatically aligned datasets achieve higher coverage than the manually aligned run.

Configurations	BLEU	NIST	METEOR	Coverage
manual	0.5222	6.8931	71.8531%	68.5417%
skip1_score1	0.5038	6.8673	71.3805%	71.8750%
skip1_score2	0.5296	6.8557	72.7302%	***72.5000%***
skip2_score1	0.5091	6.9145	71.7764%	71.8750%
skip2_score2	***0.5333***	6.8855	72.9615%	***72.5000%***
skip1_score1_span1	0.5258	6.9004	72.5916%	***72.5000%***
skip1_score2_span1	0.5285	6.8452	***73.0014%***	***72.5000%***
skip2_score1_span1	0.5273	***6.9384***	72.7157%	***72.5000%***
skip2_score2_span1	0.5290	6.8762	72.8765%	***72.5000%***

Table 2: Translation scores for the various aligner configurations.

We can make the following observations:

- the use of the *score2* scoring function gives better translation scores for the BLEU and METEOR metrics;
- switching on the **selection** *span1* alignment feature gives better METEOR scores;
- **selection** *skip2* results in better BLEU and NIST scores versus *skip1*.

Unexpectedly, the results of the extrinsic evaluation do not strictly follow the trends we found in the intrinsic evaluation. Further analysis of the data revealed that direct comparison of the manual and automatic alignments is not appropriate, especially with regards to the word-alignments. The manual alignments were produced with an aim to maximise precision, whereas we have found that our coverage-based alignments lead to higher translations. scores. This leads to having many fewer manual word-alignments than automatic ones, which in turn explains the low precision scores in the intrinsic evaluation. From this we conclude that the improvement of the automatic aligner should not be aimed at better matching the manual alignments, but rather at improving the quality of the translations produced using the automatic alignments.

5. Conclusions

Regarding aligner configurations, all evaluations indicate that including *span1* makes the most difference to alignment quality and that there's little to choose between the other configuration possibilities. Still, we do not have evidence that any particular configuration of the aligner should be preferred and recent experiments have not shown any significant differences.

Despite the clear differences between the automatic and manual alignments highlighted in the evaluation of alignment quality given in Section 4.1, we have shown that the translation scores for the automatically induced alignments are very competitive and coverage scores actually improve over the manual alignments.

It is perhaps somewhat surprising that the translation scores do not reflect the indication given by the alignment evaluation that word-level alignment precision is lower than phrase-level precision. The explanation for this may lie in how the MT system works: because DOT displays a preference for using larger fragments when building translations wherever possible, the impact of inconsistencies amongst smaller fragments (i.e. word-level alignments) is minimised.

Nevertheless, the evaluations we have presented indicate that our algorithm performs well and provides a viable solution to the challenge of inducing sub-tree alignments.

6. Future Work

Application of our alignment algorithm to parsed sections of the English–Spanish and English–German EuroParl corpora (Koehn, 2005) is currently underway. We intend to replicate the evaluations presented here for these datasets in order to (i) gain a clearer picture of differences in performance between aligner configurations and (ii) to demonstrate the language-independent nature of the alignment strategy. We are also currently investigating other uses of the automatically aligned data. One such use is the extraction of the aligned phrases for use in a phrase-based SMT system.

In addition to our current use of word-translation probability tables, we expect that factoring in phrase-table probabilities when either scoring or selecting hypothesised links will lead to increased accuracy.

Another avenue for further work centres around the adaptation of our existing algorithm to the tasks of tree-to-string, string-to-tree and string-to-string alignment, where phrase-structure will be constructed to accommodate the links that are made. We also plan to investigate the option of using *n*-best parses for the sentences and allowing our algorithm to select the best parse according to the links being induced.

Acknowledgements

This work was generously supported by Science Foundation Ireland (grant no. 05/RF/CMS064) and the Irish Centre for High-End Computing (http://www.ichec.ie). We thank Khalil Sima'an, Declan Groves and the anonymous reviewers for their insightful comments.

References

Banerjee, S. & Lavie, A. (2005). METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgements. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)* (pp. 65–72). Ann Arbor, Michigan.

Bod, R., Scha, R. & Sima'an, K. (eds.) (2003). Data-Oriented Parsing. Stanford: CSLI Publications.

Ding, Y., Gildea, D. & Palmer, M. (2003). An Algorithm for Word-Level Alignment of Parallel Dependency Trees. In *Machine Translation Summit IX* (pp. 95–101). New Orleans, LA.

Doddington, G. (2002). Automatic Evaluation of Machine Translation Quality Using N-Gram Co-Occurrence Statistics. In *Proceedings of the ARPA Workshop on Human Language Technology* (pp. 128-132). San-Diego CA.

Eisner, J. (2003). Learning Non-Isomorphic Tree Mappings for Machine Translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL-03), Companion Volume* (pp. 205–208). Sapporo, Japan.

Gildea, D. (2003). Loosely Tree-Based Alignment for Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics (ACL-03)* (pp. 80–87). Sapporo, Japan.

Groves, D., Hearne, M. & Way, A. (2004). Robust Sub-Sentential Alignment of Phrase-Structure Trees. In *Proceedings of the 20th International Conference on Computational Linguistics (CoLing 2004)* (pp. 1072-1078). Geneva, Switzerland: COLING.

Hearne, M. (2005). Data-Oriented Models of Parsing and Translation. PhD thesis. School of Computing, Dublin City University, Dublin, Ireland.

Hearne, M. & Way, A. (2003). Seeing the Wood for the Trees: Data-Oriented Translation. In *Proceedings of the MT Summit IX* (pp. 165-172). New Orleans, LA.

Hearne, M. & Way, A. (2006). Disambiguation Strategies for Data-Oriented Translation. In *Proceedings of the 11th Conference of the European Association for Machine Translation (EAMT-06)* (pp. 59–68). Oslo, Norway.

Hearne, M., Tinsley, J., Zhechev, V. & Way, A. (2007). Capturing Translational Divergences with a Statistical Tree-to-Tree Aligner. In *Proceedings of the 11th Conference on Theoretical and Methodological Issues in Machine Translation (TMI-07)*. Skövde, Sweden.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the MT Summit X* (pp. 79–86). Phuket, Thailand.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A. & Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the Demo and Poster Sessoins of the 45th Annual Meegting of the Association for Computational Linguistics (ACL-07)* (pp. 177–180). Prague, Czech Republic.

Lü, Y., Zhou, M., Li, S., Huang, C.-N. & Zhao, T. (2001). Automatic Translation Template Acquisition Based on Bilingual Structure Alignment. In *Computational Linguistics and Chinese Language Processing*, Vol. 6, No.1 (pp. 83–108). China.

Matsumoto, Y., Ishimoto, H. & Utsuro, T. (1993). Structural Matching of Parallel Texts. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL-93)* (pp. 23–30). Columbus, OH.

Nesson, R., Shieber, S. M. & Rush, A. (2006). Induction of Probabilistic Synchronous Tree-Insertion Grammars for Machine Translation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA-06)* (pp. 128–137). Boston, MA.

Och, F. J. & Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1), 19–51.

Papineni, K., Roukos, S., Ward, T. & Zhu, W.-J. (2002). BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics (ACL-02)* (pp. 311-318). Philadelphia, PA.

Poutsma, A. (2003). Machine Translation with Tree-DOP. In R. Bod, R. Scha and K. Sima'an (eds.), *Data-Oriented Parsing* (pp. 339–359). Stanford, CA: CSLI Publications.

Wang, W., Huang, J.-X., Zhou, M. & Huang, C.-N. (2002). Structure Alignment Using Bilingual Chunking. In *Proceedings of the 19th Conference on Computational Linguistics*, Vol. 1 (pp. 1–7). Taipei, Taiwan.

Wellington, B., Waxmonsky, S. & Melamed, I. D. (2006). Empirical Lower Bounds on the Complexity of Translational Equivalence. In *Proceedings of the 44th Annual Conference of the Association for Computational Linguistics (ACL)* (pp. 977-984). Sydney, Australia.

Wu, D. (1995). An Algorithm for Simultaneously Bracketing Parallel Texts by Aligning Words. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL-95)* (pp. 244–251). Cambridge, MA.

Wu, D. (1997). Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3), 377-403.