# Automatic Generation of Parallel Treebanks

**Ventsislav Zhechev**
NCLT, School of Computing
Dublin City University
Dublin, Ireland
vzhechev@computing.dcu.ie

**Andy Way**
NCLT, School of Computing
Dublin City University
Dublin, Ireland
away@computing.dcu.ie

## Abstract

The need for syntactically annotated data for use in natural language processing has increased dramatically in recent years. This is true especially for parallel treebanks, of which very few exist. The ones that exist are mainly hand-crafted and too small for reliable use in data-oriented applications. In this paper we introduce a novel platform for fast and robust automatic generation of parallel treebanks. The software we have developed based on this platform has been shown to handle large data sets. We also present evaluation results demonstrating the quality of the derived treebanks and discuss some possible modifications and improvements that can lead to even better results. We expect the presented platform to help boost research in the field of data-oriented machine translation and lead to advancements in other fields where parallel treebanks can be employed.

## 1 Introduction

In recent years much effort has been made to make use of syntactic information in statistical machine translation (MT) systems (Hearne and Way, 2006, Nesson et al., 2006). This has led to increased interest in the development of parallel treebanks as the source for such syntactic data. They consist of a parallel corpus, both sides of which have been parsed and aligned at the sub-tree level.

So far parallel treebanks have been created manually or semi-automatically. This has proven to be a laborious and time-consuming task that is prone to errors and inconsistencies (Samuelsson and Volk, 2007). Because of this, only a few parallel treebanks exist and none are of sufficient size for productive use in any statistical MT application.

In this paper we present a novel platform for the automatic generation of parallel treebanks from parallel corpora and discuss several methods for the evaluation of the results. We discuss algorithms both for cases in which monolingual parsers exist for both languages and for cases in which such parsers are not available. The parallel treebanks created with the methods described in this paper can be used by different statistical MT applications and for translation studies.

We start in section 2 by introducing the techniques for automatic generation of parallel treebanks. The evaluation methods and results are introduced in section 3 and in section 4 we give suggestions for possible improvements to the generation technology and to the evaluation algorithms. Finally, in section 5 we present existing parallel treebanks and conclude in section 6.

## 2 Automatic Generation of Parallel Treebanks

In this section we introduce a method for the automatic generation of parallel treebanks from parallel corpora. The only tool that is required besides the software presented in this paper is a word alignment tool. Such tools exist and some are freely available (eg. GIZA++ (Och and Ney, 2003)). If monolingual phrase-structure parsers[1] or at least POS taggers exist for both languages, their use for pre-processing the data is highly recommended.

In all cases, a word alignment tool is used to first obtain word-alignment probabilities for the

[1] Henceforth, we will use ‘parser’ to mean ‘monolingual phrase-structure parser’, unless stated otherwise.

parallel corpus in question for both language directions. We will start with the description of the case in which parsers are available for both languages, as this is the core of the system. The parsers are used to parse both sides of the parallel corpus. The resulting parsed data and word-alignment probability tables are then used as the input to a sub-tree alignment algorithm that introduces links between nodes in corresponding trees according to their translational equivalence scores. The output of the sub-tree aligner is the desired parallel treebank.

If there is no parser available for at least one of the languages, the parallel corpus — together with the word-alignment tables — is fed directly to a modified version of the sub-tree aligner. In this modification of the alignment algorithm, all possible binary phrase-structure trees are hypothesised for each sentence in a sentence pair. Afterwards — during the induction of alignments — only those tree nodes are left intact that take part in the alignments or are necessary for the production of connected trees. Thus, the output is again a parallel treebank with unambiguous phrase-structure trees for each language side.

In the present version of our software, if a parser or a POS tagger exists only for one of the languages in the parallel corpus you want to work with, they cannot be made use of. With the tree-to-tree and string-to-string modules in place, it is a minor task to add a tree-to-string and string-to-tree modules that will allow for the maximum utilisation of any available resources. We plan to start the development and evaluation of these new modules shortly.

We will now look at the currently available alignment algorithms in greater detail, starting with the tree-to-tree alignment and then moving on to the string-to-string case.

## 2.1 Tree-to-Tree Alignment

First, the tree-to-tree aligner has to follow certain principles to fit in the framework described above:

- Independence with respect to language pair, constituent-labelling scheme and POS tag set. Any language-dependence would require human input to adjust the aligner to a new language pair.
- Preservation of the original tree structures. We regard these structures as accurate encodings of the languages, and any change to them might distort the encoded information.
- Dependence on a minimal number of external resources, so that the aligner can be used even for languages with few available resources.
- The word-level alignments should be guided by links higher up the trees, where more context information is available.

These principles guarantee the usability of the algorithm for any language pair in many different contexts. Additionally, there are a few well-formedness criteria that have to be followed to enforce feasible alignments:

- A node in a tree may only be linked once.
- Descendants / ancestors of a source linked node may only be linked to descendants / ancestors of its target linked counterpart.

Links produced according to these criteria encode enough information to allow the inference of complex translational patterns from a parallel treebank, including some idiosyncratic translational divergences, as discussed in (Hearne et al., 2007). In what follows, a hypothesised alignment is regarded as incompatible with the existing alignments if it violates any of these criteria.

The sub-tree aligner operates on a per sentence-pair basis and each sentence-pair is processed in two stages. First, for each possible hypothetical link between two nodes, a translational equivalence score is calculated. Only the links for which a nonzero score is calculated are stored for further processing. Unary productions from the original trees, if available, are collapsed to single nodes, preserving all labels. Thus the aligner will consider a single node — instead of several nodes — for the same lexical span. This does not reduce the power of the aligner, as the translational equivalence scores are based on the surface strings and not on the tree structures.

During the second stage, the optimal combination of links is selected from among the available nonzero links. The selection can be performed using either a greedy search, or a full search for the best combination.

### Translational Equivalence

Given a tree pair $\langle S, T\rangle$ and a hypothesis $\langle s, t\rangle$, we first compute the strings in (1), where $\langle s_i \ldots s_{ix}\rangle$ and $\langle t_j \ldots t_{jy}\rangle$ denote the terminal sequences dominated by $s$ and $t$ respectively, and $\langle S_1 \ldots S_m\rangle$ and $\langle T_1 \ldots T_n\rangle$ denote the terminal sequences dominated by $S$ and $T$. Here, *inside* are the strings that represent the spans of the nodes being linked and *outside* are the strings that lay outside the spans of those nodes.

$$\begin{array}{lll} & \textit{inside} & \textit{outside} \\ (1) & s_l = \langle s_i \ldots s_{ix} \rangle & \bar{s}_l = \langle S_1 \ldots s_{i-1} s_{ix+1} \ldots S_m \rangle \\ & t_l = \langle t_j \ldots t_{jy} \rangle & \bar{t}_l = \langle T_1 \ldots t_{j-1} t_{jy+1} \ldots T_n \rangle \end{array}$$

$$(2) \quad \gamma(\langle s, t \rangle) = \alpha(s_l | t_l) \cdot \alpha(t_l | s_l) \cdot \alpha(\bar{s}_l | \bar{t}_l) \cdot \alpha(\bar{t}_l | \bar{s}_l)$$

$$(3) \quad \alpha(x|y) = \prod_i^{|x|} \sum_j^{|y|} P(x_i | y_j) \Big/ |y|$$

The score for the given hypothesis $\langle s, t \rangle$ is computed using (2) and (3). According to the formula in (3), the word-alignment probabilities are used to get an average vote by the source tokens for each target token. Then the product of the votes for the target words gives the alignment probability for the two strings. The final translational equivalence score is the product of the alignment probabilities for the inside and outside strings in both language directions as in (2).

## Greedy-Search Algorithm

The greedy-search algorithm is very simple. The set of nonzero-scoring links is processed iteratively by linking the highest-scoring hypothesis at each iteration and discarding all hypotheses that are incompatible with it until the set is empty.

Problems arise when there happen to be several hypotheses that share the same highest score. There are two distinct cases that can be observed here: these top-scoring hypotheses may or may not represent incompatible links. If all such hypotheses are compatible, they are all linked at the same time and all remaining unprocessed hypotheses that are incompatible with any of those links are discarded. In case even one among the top-scoring hypotheses is incompatible with the others, these hypotheses are skipped and processed at a later stage.

The sub-tree aligner can be built to use one of two possible skipping strategies, which we will call *skip1* and *skip2*. According to the *skip1* strategy, hypotheses are simply skipped until a score is reached, for which only one hypothesis exists. This hypothesis is then linked and the selection algorithm continues as usual.

The *skip2* strategy is more complex, in that we also keep track of which nodes take part in the skipped hypotheses. Then, when a candidate for linking is found, it is only linked if it does not include any of these nodes. The motivation behind this strategy is that a situation may occur 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.

Regardless of whether *skip1* or *skip2* is used, sometimes a situation occurs in which the only hypotheses remaining unprocessed are equally likely candidates for linking according to the selection strategy. In such ambiguous cases our decision is not to link anything, rather than make a decision that might be wrong.

During initial testing of the aligner we found that often lexical links would get higher scores than the non-lexical links,[2] which sometimes resulted in poor lexical links preventing the selection of bona fide non-lexical ones. To address this issue, an extension to the selection algorithm was developed, which we call *span1*. When enabled, this extension results in the set of nonzero hypotheses being split in two subsets: one containing all hypotheses for lexical links, and one containing the hypotheses for non-lexical links. Links are then first selected from the second subset, and only when it is exhausted does the selection continue with the lexical one. This division does not affect the discarding of incompatible links after linking; incompatible links are discarded in whichever set they are found.

## Full-Search Algorithm

This is a backtracking recursive algorithm that enumerates all possible combinations of non-crossing links. All maximal combinations[3] found during the search are stored for further processing. After the search is complete, the probability mass of each combination is calculated by summing the translational equivalence scores for all the links in the combination. The maximal combination of non-crossing links that has the highest probability mass is selected as the best alignment for the sentence pair.

Often, there are several distinct maximal combinations that share the highest probability mass; for longer sentences this number can rise to several hundred. The disambiguation strategy that we currently employ is to take the largest common subset of all maximal combinations. Another strategy would be to output all possible combinations and mark them as relating to the same sentence pair, thus leaving the disambiguation to the application that uses the resulting parallel treebank.

[2] *lexical* are such links, for which at least one of the linked nodes spans over only one word. All other links are *non-lexical*.

[3] A maximal combination of non-crossing links is a combination of links for which any newly added link would be incompatible with at least one of the links already in the combination.

### 2.2 String-to-String Alignment

The string-to-string aligner can accept as its input plain or POS-tagged data. For a pair of sentences, all possible binary trees are first constructed for each sentence. All nodes in these trees have the same label (*X*) and are then used as available link targets. In the case of POS-tagged data, the pre-terminal nodes receive the POS tags as labels. Here it is obvious that the number of links will be much higher than for the sub-tree alignment case, so the string-to-string aligner will operate much more slowly.

After all link-hypothesis scores have been calculated, the string-to-string aligner continues with the selection of links in the same manner as the sub-tree aligner, with one extension; after a link has been selected — besides all incompatible links — all binary trees that do not include the linked nodes are discarded with any nonzero hypotheses attached to them. In this way, only those binary trees that are compatible with the selected links remain after the linking process.

In an additional step for the string-to-string aligner, all non-linked nodes (except for the root nodes) are discarded, thus allowing for the construction of unambiguous *n*-ary trees for the source and target sentences. If necessary, non-linked nodes are left intact to provide supporting structure in the trees.

## 3 Evaluation and Results

The quality of a parallel treebank depends directly on the quality of the sub-tree alignments that it contains. Because of this, we use the evaluation results mainly as a metric for the improvements in the sub-tree aligner during development. Of course, the evaluation presented in this section also presents an insight into the usability of the parallel treebanks produced using our method.

For the evaluation of the aligner, a battery of intrinsic and extrinsic tests was developed. As a reference for the tests, a hand-crafted parallel treebank was used (HomeCentre (Hearne and Way, 2006)). This treebank consists of 810 English–French sentence pairs. As discussed in section 5, we are not aware of an existing parallel treebank besides the HomeCentre that can be used directly for cross evaluation and comparison to versions automatically generated using the sub-tree aligner.

The word-alignment probabilities required by our system were obtained by running the Moses decoder[4] (Koehn et al., 2007) on the plain sentences from the HomeCentre in both language directions.

We will first describe the intrinsic testing and then go into the details of the extrinsic evaluation.

### 3.1 Intrinsic Evaluation

The intrinsic evaluation is performed by comparing the links induced by the automatic aligner to the manually annotated links in the HomeCentre treebank. This evaluation can only be performed for the result of the tree-to-tree alignment, as the string-to-string alignment produces different trees. The metrics used for the comparison are precision and recall for all alignments and lexical and non-lexical alignments alone. The results of the evaluation are shown in Table 1.[5]

| | all links | | lexical links | | non-lexical links | |
|---|---|---|---|---|---|---|
| Configurations | precision | recall | precision | recall | precision | recall |
| *skip1* | *61,29%* | *77,46%* | *51,06%* | *79,99%* | **80,75%** | *75,69%* |
| *skip2* | 61,54% | 77,50% | 51,29% | 80,03% | **80,75%** | 75,70% |
| *skip1_span1* | 61,56% | 78,44% | 51,53% | 80,51% | *78,67%* | **77,22%** |
| *skip2_span1* | **61,79%** | **78,49%** | **51,76%** | **80,60%** | 78,73% | **77,22%** |

Table 1. Intrinsic evaluation results

Looking first to the *all links* column, it is immediately apparent that recall is significantly higher than precision for all configurations. In fact, all aligner variations consistently induce on average two more links than exist in the manual version. Considering the *lexical links* and *non-lexical links* columns, apparently the bulk of the automatically induced links that do not occur in the manual annotation are at the lexical level, as attested by the low precision at the lexical level and balanced precision and recall at the non-lexical level.

If the manual alignments in the HomeCentre are regarded as a gold standard, it would seem that fewer lexical links should be produced, while the quality of the non-lexical links needs improvement. We will try to judge whether this is really the case using the extrinsic evaluation techniques described below.

### 3.2 Extrinsic Evaluation

For extrinsic evaluation, we trained and tested a DOT system (Hearne and Way, 2006) using the manually aligned HomeCentre treebank and evaluated the output translations to acquire baseline scores. We then trained the system on the automatically generated treebank and repeated

[4] We found that using the Moses word-alignment probabilities yielded better results than those output directly by GIZA++.

[5] Throughout the paper we use **boldface** to highlight the best results and *italics* for the worst.

the same tests, such that the only difference across runs are the alignments.

For testing, we used the six English–French training / test splits for the HomeCentre used in (Hearne and Way, 2006). Each test set contains 80 test sentences and each training set contains 730 tree pairs. We evaluated the translation output using three automatic evaluation metrics: BLEU (Papineni et al., 2002), NIST (Doddington, 2002) and METEOR (Banerjee and Lavie, 2005). We averaged the results over the six splits. We also measured test-data coverage of the translation system, i.e. the percentage of test sentences for which full trees were generated during translation.

We performed this evaluation using both the tree-to-tree algorithm and the string-to-string algorithm, employing greedy-search selection. For the latter case we extracted POS-tagged sentences from the HomeCentre and used them as input for the aligner. The results for the tree-to-tree case are presented in Table 2 and for the string-to-string case in Table 3.

| Configurations | BLEU | NIST | METEOR | Coverage |
|---|---|---|---|---|
| *manual* | *0,5222* | 6,8931 | *71,8531%* | *68,5417%* |
| *skip1* | 0,5236 | *6,8412* | 72,2485% | **72,0833%** |
| *skip2* | 0,5233 | 6,8617 | 72,2847% | 71,8750% |
| *skip1_span1* | 0,5296 | 6,8570 | **72,9833%** | **72,0833%** |
| *skip2_span1* | **0,5334** | **6,9210** | 72,9736% | 71,8750% |

Table 2. Tree-to-tree extrinsic evaluation

Let us first look at the results from the tree-to-tree aligner. Overall, the scores obtained when using the manual alignments are very competitive with those derived using the manually aligned data. In fact, NIST is the only metric for which the performance is below the baseline. An important observation is that the coverage of the translation system is up to 3.5% higher when using the automatic alignments. Another observation is that *skip2* leads to better performance on the NIST metric over *skip1*, but the results from the other metrics are not so conclusive. The use of *span1* leads to better translation scores. These results seem to point at the *skip1_span1* and *skip2_span1* configurations as the best-suited for further development.

Unexpectedly, the results of the extrinsic evaluation do not strictly follow the trends found in the intrinsic evaluation. Further analysis of the data revealed that direct comparison of the manual and automatic alignments is not appropriate, especially regarding the lexical alignments. The manual alignments were produced with the aim of maximising precision, but the coverage-based automatic alignments lead to higher translation scores. This is the result of having many fewer manual word-alignments than automatic ones, as the low precision scores in the intrinsic evaluation show. 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.

| Configurations | BLEU | NIST | METEOR | Coverage |
|---|---|---|---|---|
| *manual* | **0,5222** | **6,8931** | *71,8531%* | *68,5417%* |
| *skip1* | 0,4939 | 6,6321 | 72,5192% | **92,5000%** |
| *skip2* | 0,4886 | 6,5777 | 72,8241% | 92,2917% |
| *skip1_span1* | *0,4661* | *6,3090* | 73,1017% | 92,2917% |
| *skip2_span1* | 0,4683 | 6,3353 | **73,2828%** | 92,2917% |

Table 3. String-to-string extrinsic evaluation

If we now look at the evaluation of the string-to-string aligner, we see quite peculiar results. There is more than 20% increase in coverage compared to the tree-to-tree aligner, but the only other metric that sees improvement — albeit modest — is METEOR. It is also the only metric that follows the trends observed in the tree-to-tree evaluation results. Not only are the results for the BLEU and NIST metrics lower, but they also seem to follow reversed trends. It is unclear what the reason for such an outcome is, and further investigation — including on other data sets — is needed. Still, as far as the METEOR metric is concerned, the use of the string-to-string algorithm for the generation of parallel treebanks seems to be warranted.

The results obtained from the intrinsic and extrinsic evaluations show that the methods described in this paper produce high quality parallel treebanks. Using the automatically generated treebanks, a DOT system produces results with similar translation quality and better coverage compared to its performance using manually aligned data. This makes our methods a good alternative to the manual construction of parallel treebanks.

### 3.3 Using the Full-Search Algorithm as an Evaluation Metric

The full-search selection algorithm is combinatorial in nature and for sentence pairs with more than 100 nonzero link hypotheses its time requirements become prohibitive. Still, this algorithm can be used in its current form for development purposes.

It is reasonable to ask whether the greedy-search algorithm produces the best set of alignments for a given sentence pair. It could be that it picks a local maximum differing greatly from the absolute maximal set of alignments, thus producing either low quality links or a small number of links.

The full-search selection algorithm can be used to test the performance of the greedy search, as it by definition produces the best available set of alignments. We decided to use the rate of coincidence between the alignments induced using both selection algorithms as a metric for the quality of the links derived using the greedy search: the higher the number of cases in which the greedy-search algorithm matches the result of the full-search algorithm, the better the quality of the greedy search.

We ran this coincidence evaluation for all four configurations of the aligner. The results are presented in Table 4. It should be noted that 30 sentence pairs from the HomeCentre could not be handled by the full-search algorithm within a reasonable timeframe and were skipped.

| | all links | | lexical links | | non-lexical links | |
|---|---|---|---|---|---|---|
| Configurations | precision | recall | precision | recall | precision | recall |
| *skip1* | 98,71% | 99,18% | 98,36% | 99,14% | **99,57%** | *99,21%* |
| *skip2* | **99,23%** | **99,21%** | **99,06%** | **99,17%** | **99,57%** | **99,23%** |
| *skip1_span1* | *95.78%* | *97,00%* | *95.92%* | *96,33%* | *95.19%* | *99,21%* |
| *skip2_span1* | 96,27% | 97,09% | 96,58% | 96,44% | 95,25% | *99,21%* |

Table 4. Evaluation against full-search results

The outcome of this test seems to be unexpected and a little disconcerting in view of the results obtained from the extrinsic evaluation. It does not seem reasonable that the configurations including *span1* should obtain scores that are relatively much worse than the scores for the other configurations, when we saw them perform better at the extrinsic evaluation tests.

The reason for this discrepancy might not be obvious, but it is fairly simple and lies in the nature of the *span1* extension. As discussed in section 2.1, *span1* introduces a separation in the induction of lexical and non-lexical links. The full-search algorithm, however, derives the maximal link set from a common pool of all nonzero alignment hypotheses. This suggests that an extension to the full-search algorithm similar to *span1* should be developed to allow for the evaluation of configurations using this feature.

Nevertheless, this evaluation shows some very important results. Besides the fact that configurations using *skip2* perform slightly better than those using *skip1*, we see that the greedy search comes very close to the best maximal link set. Our tests show that in over 95% of the cases the greedy search finds the best maximal link set available for the particular sentence pair.

The results are very encouraging and show that the fast greedy-search algorithm produces the desired results and there is no need to use the prohibitively slow full-search algorithm, except for comparison purposes.

## 4 A Review of Possible Enhancements

Here, we discuss possible avenues for the improvement of the quality of the parallel treebanks produced using the methods presented in this paper.

As already stated in section 3, the quality of a parallel treebank is to be judged by the quality of the induced sub-tree alignments. Thus, all effort should be directed at producing better alignments. There are two possible ways to address this: one option is to work on improving the alignment algorithm, and the other option is to improve the scoring mechanism used by the aligner.

Improvements to the alignment algorithm can be evaluated against the full-search selection algorithm. The evaluation results from section 3.3 suggest, however, that the margin for improvement here is very small. Thus, we do not expect any improvements here to bring serious boosts in overall performance. Nevertheless, we plan to investigate one possible modification to the greedy search.

It can be argued that each newly induced link in a sentence pair should affect the decisions regarding which links to select further in the alignment process for this sentence pair. This can be simulated to a certain extent by the introduction of a simple re-scoring module to the aligner. Each time a new link has been selected, this module will be used to recalculate the scores of the remaining links, considering the restrictions on the possible word-level alignments introduced by this link, e.g. that words within the spans of the nodes being linked cannot be aligned to words outside those spans.

The effects of changes to the scoring mechanism used can only be evaluated using extrinsic methods, as such changes also influence the operation of the full-search selection. On this front, we plan to investigate a maximum-entropy-based scoring mechanism. We expect such a mechanism to better encode mathematically the dependence of the translational equivalence scores on the word-alignment probabilities.

Besides the improvements to the sub-tree aligner, we plan to extend the whole generation framework with two additional modules: for string-to-tree and tree-to-string alignment. This would allow for better utilisation of all available resources for the derivation of a parallel treebank from a parallel corpus.

We also plan to perform large-scale extrinsic evaluation experiments. Though the evaluation results presented in section 3 are very promising, they were performed on a very small set of data. (John Tinsley (p.c.) reports successfully deriving a parallel treebank with over 700 000 sentence-pairs using our software.) Further experiments on larger data sets — from different languages, as well as from different domains — should help better understand the real qualities of the methods presented here.

## 5 Existing Parallel Treebanks

In this section we look at several attempts at the creation of parallel treebanks besides the Home-Centre treebank presented earlier.

Closest to the material presented in this paper comes the parallel treebank presented in (Samuelsson and Volk, 2006). This manually created treebank aligns three languages — German, English and Swedish — consisting of over 1000 sentences from each language. The main difference compared to our method is that they allow many-to-many lexical alignments and one-to-many non-lexical alignments. The authors also allow unary productions in the trees, which, as stated in section 2.1, does not provide any additional useful information. Another difference is that they deepen the original German and Swedish trees before alignment, rather than preserve their original form.

A further attempt to align phrase-structure trees is presented in (Uibo et al., 2005). The authors develop a rule-based method for aligning Estonian and German sentences. The parallel treebank consist of over 500 sentences, but in the version presented only NPs are aligned.

In (Han et al., 2002) the authors claim to have built a Korean–English parallel treebank with over 5000 phrase-structure tree pairs, but at the time of writing we were unable to find details about this treebank.

Although the Prague Czech–English Dependency Treebank (PCEDT (Čmejrek et al., 2004)) can be used as a parallel treebank, it is not such per se. The authors do not use phrase-structure trees. Instead, tectogrammatical dependency structures are used (Hajičová, 2000). Either a word alignment tool like GIZA++ or a probabilistic electronic dictionary (supplied with the treebank) can be used to automatically align the dependency structures. The presented version contains over 21000 sentence pairs that can be aligned. Because of its nature, this treebank can only be used by MT systems that employ tectogrammatical dependency structures.

We are also aware of the existence of the LinES (Ahrenberg, 2007), CroCo (Hansen-Schirra et al., 2006) and FuSe (Cyrus, 2006) parallel corpora. Although it seems possible to use them as parallel treebanks, they have been designed to serve as resources for the study of translational phenomena and it does not appear that they can be used effectively for other natural language processing tasks.

An attempt to develop an automatic tree-to-tree aligner is described in (Groves et al., 2004). The authors present a promising rule-based system. Further testing, however, has shown that the rules are only applicable to a particular treebank and language pair. This means that the set of rules has to be adjusted for each particular case.

Thus, the methods presented in this paper are the only available ones that can be used to produce a sufficiently large parallel treebank appropriate for use by state-of-the-art statistical MT applications (eg. DOT (Hearne and Way, 2006)).[6]

## 6 Conclusions

We have presented a novel platform for the fast and robust automatic generation of parallel treebanks. The algorithms described are completely language-pair-independent and require a minimal number of resources; besides a parallel corpus, a word alignment tool is the only extra software required. If available, POS taggers or monolingual phrase-structure parsers can be used to preprocess the data. Certain extensions to the current software are planned that will assure the optimal use of any available resources.

A series of evaluations have shown promising results. The quality of the automatically generated parallel treebanks is very high, even improving on a manually created treebank on certain metrics. We plan to carry out extensive large-scale testing on a range of language pairs, which we expect to corroborate the results reported in this paper. The planned improvements to the algorithms discussed in section 4 are expected to further increase the quality of the generated parallel treebanks.

Currently existing treebanks are small and require extensive human resources to be created and extended, which has limited their use for data-oriented tasks. The platform presented in this paper provides a means to circumvent these problems by allowing for the fast automatic generation of very large parallel treebanks with very little human effort, thus overcoming this hurdle for research in tree-based machine translation.

[6] An alternative methodology is described in (Lavie et al., to appear), but this work was not available at the time of writing.

## Acknowledgements

We would like to thank Khalil Sima'an, Mary Hearne and John Tinsley for many insightful discussions. 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).

## References

Ahrenberg, Lars. 2007. LinES: An English-Swedish Parallel Treebank. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA '07)*, pp. 270–274. Tartu, Estonia.

Banerjee, Satanjeev and Alon Lavie. 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, MI.

Čmejrek, Martin, Jan Cuřín, Jiří Havelka, Jan Hajič and Vladislav Kuboň. 2004. Prague Czech-English Dependency Treebank: Syntactically Annotated Resources for Machine Translation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC '04)*. Lisbon, Portugal.

Cyrus, Lea. 2006. Building a resource for studying translation shifts. In *Proceedings of the 5th Conference of Language Resources and Evaluation (LREC '06)*, pp. 1240–1245. Genoa, Italy.

Doddington, George. 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.

Groves, Declan, Mary Hearne and Andy Way. 2004. Robust Sub-Sentential Alignment of Phrase-Structure Trees. In *Proceedings of the 20th International Conference on Computational Linguistics (CoLing '04)*, pp. 1072–1078. Geneva, Switzerland: COLING.

Hajičová, Eva. 2000. Dependency-Based Underlying-Structure Tagging of a Very Large Czech Corpus. *TAL (Special Issue Grammaires de Dépendance / Dependency Grammars)*, 41 (1): 47–66.

Han, Chung-hye, Na-Rare Han, Eon-Suk Ko and Martha Palmer. 2002. Development and Evaluation of a Korean Treebank and its Application to NLP. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC '02)*, pp. 1635–1642. Las Palmas, Canary Islands, Spain.

Hansen-Schirra, Silvia, Stella Neumann and Mihaela Vela. 2006. Multi-dimensional Annotation and Alignment in an English-German Translation Corpus. In *Proceedings of the workshop on Multi-dimensional Markup in Natural Language Processing (NLPXML '06)*, pp. 35–42. Trento, Italy.

Hearne, Mary and Andy Way. 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, Mary, John Tinsley, Ventsislav Zhechev and Andy Way. 2007. Capturing Translational Divergences with a Statistical Tree-to-Tree Aligner. In *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI '07)*, pp. 85–94. Skövde, Sweden: Skövde University Studies in Informatics.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the Demo and Poster Sessions of the 45th Annual Meeting of the Association for Computational Linguistics (ACL '07)*, pp. 177–180. Prague, Czech Republic.

Lavie, Alon, Alok Parlikar and Vamshi Ambati. to appear. Syntax-driven Learning of Sub-sentential Translation Equivalents and Translation Rules from Parsed Parallel Corpora. In *Proceedings of the 2nd Workshop on Syntax and Structure in Statistical Translation (SSST '08)*. Columbus, OH.

Nesson, Rebecca, Stuart M. Shieber and Alexander Rush. 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, Franz Josef and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29 (1): 19–51.

Papineni, Kishore, Salim Roukos, Todd Ward and Wei-Jing Zhu. 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.

Samuelsson, Yvonne and Martin Volk. 2006. Phrase Alignment in Parallel Treebanks. In *Proceedings of the 5th Workshop on Treebanks and Linguistic Theories (TLT '06)*, pp. 91–102. Prague, Czech Republic.

Samuelsson, Yvonne and Martin Volk. 2007. Alignment Tools for Parallel Treebanks. In *Proceedings of the GLDV Frühjahrstaggung*. Tübingen, Germany.

Uibo, Heli, Krista Liin and Martin Volk. 2005. Phrase alignment of Estonian-German parallel treebanks. Paper presented at *Workshop 'Exploiting parallel corpora in up to 20 languages'*, Arona, Italy.