

Large Aligned Treebanks for Syntax-based Machine Translation

Gideon Kotzé, Vincent Vandeghinste, Scott Martens, Jörg Tiedemann

University of Groningen, University of Leuven, University of Tübingen, University of Uppsala
g.j.kotze@rug.nl, vincent@ccl.kuleuven.be, scott.martens@uni-tuebingen.de, jorg.tiedemann@lingfil.uu.se

Abstract

We present a collection of parallel treebanks that have been automatically aligned on both the terminal and the nonterminal constituent level for use in syntax-based machine translation. We describe how they were constructed and applied to a syntax- and example-based machine translation system called Parse and Corpus-Based Machine Translation (PaCo-MT). For the language pair Dutch to English, we present evaluation scores of both the nonterminal constituent alignments and the MT system itself, and in the latter case, compare them with those of Moses, a current state-of-the-art statistical MT system, when trained on the same data.

Keywords: Parallel Treebank, Syntax-based MT, Constituent Alignment

1. Introduction

For statistical modeling in NLP, such as that used in machine translation, large parallel corpora are typically required. Most of the data-driven MT research has been performed on flat sequence representations of sentences, on strings. By using *tree structures* one can exploit the linguistic relationships between words and phrases. This requires syntactically parsing these corpora and aligning them on the constituent level, producing a collection of parallel treebanks.

Various systems have been developed for non-terminal constituent alignment, either making use of hand-crafted rules (Menezes and Richardson, 2001; Groves et al., 2004) or fully unsupervised methods (Zhechev and Way, 2008; Lavie et al., 2008). We have opted to use *Lingua-Align*, a supervised tree-to-tree aligner implementing discriminative models (Tiedemann, 2010). We have also implemented a recently developed rule-based algorithm to produce an additional *high recall* version of our parallel treebanks, with many more alignments added. Clean corpora are tokenized, sentence aligned, parsed, word aligned and tree aligned. The end result is two fully annotated and aligned sets of parallel treebanks consisting of millions of sentence pairs in four language combinations.

In current *tree-based* MT research, most systems use one or another form of probabilistic *synchronous context-free grammars* (SCFGs) such as the tree-based models of the Moses toolkit (Koehn et al., 2007).¹ There are other, more complex types of *synchronous* grammars which can also be learned from parallel treebanks and which accept trees on both the left and right hand side of the grammar rules, allowing such operations as raising and lowering of nodes (Chiang, 2006). Depending on the operations that are allowed, we distinguish between *synchronous tree substitution grammars* (STSGs) (Schabes, 1990; Eisner, 2003) which allow substitution, *synchronous tree insertion grammars* (STIGs) (Nesson et al., 2006), which allow both substitution and insertion; and *synchronous tree adjoining grammars* (STAG) (Shieber and Schabes, 1990) which allow substitution, insertion and adjunction.

We used the parallel treebank, with alignments at the con-

stituent level to train the PaCo-MT system, which is a syntax-based MT engine using an STSG transducer to bridge the gap between source sentence parse tree and target language parse tree.

The remainder of the paper is organised as follows: Section 2. describes how the parallel treebank was constructed. Section 3. describes how we used this treebank as training data for the PaCo-MT system, and section 4. draws conclusions and describes future work.

2. Producing a parallel treebank

The parallel treebanks were produced by obtaining a collection of parallel corpora containing only the raw text and processed automatically by applying a series of tools. In the next few sections, we describe in more detail how the data was processed.

2.1. Data collection

We collected Dutch-English and Dutch-French subsets of the following parallel corpora:

- Europarl 3 (Koehn, 2005), a corpus containing the proceedings of the European Parliament in several languages from 1996 to 2006.
- the DGT Multilingual Translation Memory of the Acquis Communautaire: DGT-TM.²
- OPUS: the open parallel corpus (Tiedemann, 2009).³
- an additional private translation memory (Transmem).

2.2. Parsing

The corpora of all three languages involved were annotated using separate monolingual parsers. We first parse all the texts before they are sentence aligned. In doing so, we have the maximum amount of trees available for target language modeling.

The Dutch data was parsed using the Alpino parser (van Noord, 2006). Some additional processing was applied in order to make the resulting parse trees more uniform, for

¹<http://www.statmt.org/moses/>

²<http://langtech.jrc.it/DGT-TM.html>

³<http://opus.lingfil.uu.se/>

example by adding unary branching, aiming at a reduction in the number of patterns that occur. When Dutch data is source language data, we removed nodes with separable verb particles, and added these particles to the verb token and lemma. When Dutch is the target language, no such action was taken.

The English data was parsed using the Stanford parser (Klein and Manning 2003) with dependency information (de Marneffe and Manning, 2006).

We used the Berkeley parser (Petrov and Klein 2007) on the French data. As the model does not add dependency information, we implemented head-finding rules in a similar style as (de Marneffe and Manning, 2006).

Tables 1 and 2 display word and sentence pair counts of the 1:1 parallel treebanks.

2.3. Alignment

After parsing, the texts were extracted and sentence aligned, after which they were matched with the parse trees, resulting in a fully one-to-one mapping of parallel trees. The sentence aligner of choice was Hunalign (Varga et al., 2005).

Next, we processed the Dutch trees as mentioned above. As a result, we have four different sets of parallel data, one for each language pair and direction.

After producing the final set of one-to-one sentence pairs, we proceeded to align the tokens. For this, we utilized GIZA++ (Och and Ney, 2003), using the *intersect* and *grow-diag* heuristics implemented by Moses (Koehn et al., 2007) to produce a higher recall set of alignments suitable for MT.

The next step is aligning the nonterminal constituents. Using the Stockholm TreeAligner (Lundborg et al., 2007), we constructed a set of parallel alignments for each language pair which functions as training data for the Lingua-Align models. To achieve this, we imported the already existing word alignments for that particular set, and manually drew links between the nonterminal nodes of all parallel sentences, after which we trained our models and applied them on the treebanks. Similar to what is used in word alignment (Och and Ney, 2003) we make a distinction between "sure" (or "good") and "possible" links, where the former are links which indicate exact equivalence, and the latter include both sure and so-called "fuzzy" links, which are links which indicate an approximate equivalence that are still considered useful for translational purposes.

We consider the alignments produced by Lingua-Align our *high precision* data set. Although the models were tuned to maximize balanced F-score, recall was consistently lower than precision. Experiments suggested that linking well-formed first-level subtrees sharing word alignments is generally accurate, and that the order of alignment matters (Kotzé, 2011a; Kotzé, 2011b). This led us to develop a bottom-up rule addition module that greedily links candidate subtree pairs using measures of similarity and the existence of word alignments and with more relaxed well-formedness constraints, with the intention of implementing it on data sets that are already aligned by Lingua-Align.

The algorithm states that all source tree nonterminal nodes are traversed, starting on the first level above the terminals,

and, for every unlinked node considered, we check all unlinked candidate target tree nonterminals, starting on the same level as the source tree node. If conditions are satisfied, a link is made before moving on to another subtree pair. We recognize that traversing the target nodes first also seems reasonable, but have not yet experimented with directionality (intersect, union, and so on).

For a link to be made, all of the following criteria need to be met:

- The subtree pair must be well-formed. However, near well-formedness is allowed as long as there are not any good links to the outside of the trees and there is no more than one fuzzy link to the outside. Punctuation is ignored.
- The subtree pair must have a certain degree of similarity. In this case, we define a good enough similarity as one where both the leaf count similarity and leaf link similarity scores (defined below) obtain values above a certain threshold.

The leaf count similarity score is based on the ratio between the number of leaves dominated by the source tree node and the number of leaves dominated by the target tree node. Similarly, the leaf link similarity score is based on the ratio between the number of linked leaves and the total number of leaves dominated by the source and target side subtrees. For the leaf count similarity, the ratio, as in (Tiedemann and Kotzé, 2009), is calculated as follows:

$$leafratio(s_i, t_j) = \frac{\min(|s_x \leq s_i|, |t_y \leq t_j|)}{\max(|s_x \leq s_i|, |t_y \leq t_j|)}$$

where s_i and t_j are the respective candidate source and target side nodes, and s_x are the leaves dominated by s_i and t_y are dominated by t_j .

The linked leaf ratio is calculated as follows:

$$llr(s_i, t_j) = \frac{\sum_{s_x} link(s_x, s_i) + \sum_{t_y} link(t_y, t_j)}{\sum_{s_x} leaf(s_x, s_i) + \sum_{t_y} leaf(t_y, t_j)}$$

$$leaf(s_x, s_i) = \begin{cases} 1 & \text{if } s_x \leq s_i \\ 0 & \text{otherwise} \end{cases}$$

$$leaf(t_y, t_j) = \begin{cases} 1 & \text{if } t_y \leq t_j \\ 0 & \text{otherwise} \end{cases}$$

$$link(s_x, s_i) = \begin{cases} 1 & \text{if } s_x \leq s_i \text{ and } \exists l_{s_x} \in L \\ 0 & \text{otherwise} \end{cases}$$

$$link(t_y, t_j) = \begin{cases} 1 & \text{if } t_y \leq t_j \text{ and } \exists l_{t_y} \in L \\ 0 & \text{otherwise} \end{cases}$$

where l denotes a linked leaf and L is the set of all linked leaves (aligned terminal nodes).

We found that the ratio should play a more important role when the differences as explained above are greater. For example, it is quite common that a subtree governing a single terminal node should be linked to one that governs two terminal nodes, but it seems less likely that a subtree governing three terminals should be linked to one with six, even

Table 1: Word and sentence counts on the 1:1 Dutch/English and English/Dutch parallel treebanks.

Corpus	Dutch-English		English-Dutch	
	Sentence pairs	Words	Sentence pairs	Words
Europarl	1,180,706	57,812,824	1,190,501	57,810,789
DGT-TM	478,972	19,974,703	484,186	20,222,021
OPUS+Transmem	1,097,834	20,393,721	1,102,958	20,493,131
Total	2,757,512	98,181,248	2,777,645	98,525,941

Table 2: Word and sentence counts on the 1:1 Dutch/French and French/Dutch parallel treebanks.

Corpus	Dutch-French		French-Dutch	
	Sentence pairs	Words	Sentence pairs	Words
Europarl	1,188,022	60,987,015	1,188,757	61,231,725
DGT-TM	385,317	17,750,535	385,644	17,801,831
OPUS+Transmem	720,487	18,962,966	722,494	19,038,759
Total	2,293,826	97,700,516	2,296,895	98,072,315

though the ratios are the same. Manual investigation would suggest that the same principle can be applied to linked leaf ratios. We therefore introduce a penalty score to be subtracted from the ratios, the results of which constitute the leaf ratio similarity and linked leaf ratio similarity scores respectively.

Simply subtracting the differences would lead to a score below 0 in all cases except when the difference itself is 0. Therefore, it seemed to be in order to normalize it against an imagined maximum number of leaves. In this case, we proceeded with trial and error, and set the denominator at the value of 80.

We can now define our leaf ratio similarity score as follows:

$$lrs(s_i, t_j) = leafratio(s_i, t_j) - \frac{abs(|s_x \leq s_i| - |t_y \leq t_j|)}{z}$$

where z denotes the normalization value. Similarly, our linked leaf similarity score can be expressed as follows:

$$\begin{aligned} llr(s_i, t_j) &= llr(s_i, t_j) \\ &= \frac{|s_x \leq s_i| + |t_y \leq t_j| - (|s_x \leq s_i| + |t_y \leq t_j|)}{z} \end{aligned}$$

Provisionally, we set our thresholds for the leaf ratio similarity at 0.35 and for the linked leaf ratio similarity at 0.45. Additionally, a link is made fuzzy if its value came within the range of 0.05 of the thresholds. Finally, we assume that sentence alignment is perfect and therefore change all fuzzy links between root nodes to good.

We found that the parsers treated punctuation differently, and that the existence of a punctuation mark inside or outside a subtree should not dictate an alignment decision as much as for example a content word does. Another observation is that in many cases, subtrees should still be aligned even if a fuzzy link goes to the outside - in other words,

when a token in one of the two subtrees links to a token in another subtree. However, if a good link goes to the outside, it is less likely that they should be linked. Therefore, we have decided to allow for a certain amount of non-wellformedness in the case of fuzzy links.

Table 3 presents the evaluation results of both our high precision and high recall approaches on a gold standard of 200 sentence pairs for Dutch to English, taking only nonterminal alignments into account. The sentences were extracted from Europarl and manually aligned using the Stockholm TreeAligner. F-scores are balanced and are based on both sure and possible links.

Table 3: Evaluation scores of *high precision* vs. *high recall* approaches tested on a Dutch to English gold standard.

Data set	Precision	Recall	F-score
High precision	93.3	61.2	73.9
High recall	72.0	82.3	76.8

Noticeable is the very significant decrease in alignment precision and a similar increase in recall. The F-score, though, increases by almost 3%. Implementing our algorithm on the full data set results in the statistics presented in table 4. In section 3.3. we discuss the effects of training our system on either set.

3. Using the parallel treebanks in MT

The preprocessing steps described in the previous section allow to extract transduction rules: synchronous grammar rules indicating the mapping of the source language trees onto the target language trees. By keeping track of how many times such mappings occur, we can add weights to the synchronous grammar.

In the next section we describe how we extracted an STSG from the parallel treebanks. An extraction of SCFGs, STIGs and STAGs is also possible.

Table 4: Counts of alignments applied to all the parallel treebanks

Language pair	Moses/GIZA++	Lingua-Align	Rules	Total
Dutch/English	48,011,840	5,049,496	9,818,623	62,879,959
English/Dutch	38,060,911	11,820,036	9,301,941	58,182,888
Dutch/French	46,976,046	9,128,714	6,021,967	62,126,727
French/Dutch	47,017,307	6,237,574	5,259,590	58,514,471
Total	180,066,104	32,235,820	30,402,121	242,704,045

3.1. Rule extraction

In the extraction of an STSG for our MT system, we limit ourselves to grammar rules with a *horizontally complete* subtree on both the source and the target side of each grammar rule. Subtrees are *horizontally complete* (Boitet and Tomokiyo, 1995) if, when a daughter node of a node is included in the subtree, then so are all of its sisters.

Figure 1 is an example⁴ of two sentences aligned at both the sentence and constituent level. For each alignment point, either one or two rules are extracted. First, each alignment point is a lexical alignment, creating a rule that maps a source language word or phrase to a target language one (Figure 2 (a), (b) and (c)).

Figure 1: Two sentences with subsentential alignment.

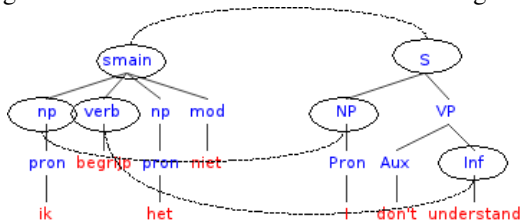
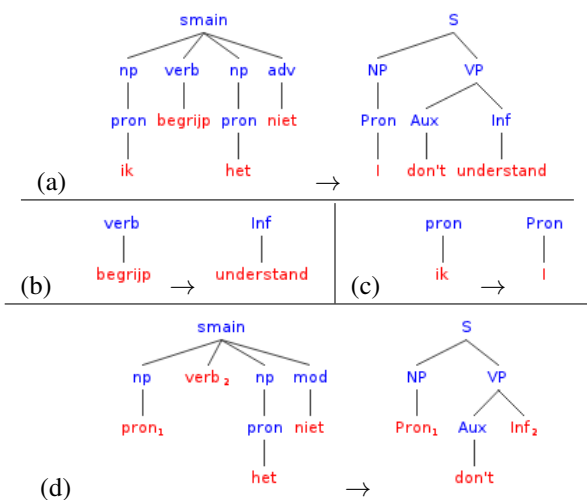


Figure 2: Rules extracted from the alignments in Figure 1



Secondly, each aligned pair of sentences engenders further rules by partitioning each tree at each alignment point, yielding non-lexical grammar rules. For these rules, the

⁴The edge labels have been omitted from these examples, but were used in the actual rule induction.

alignment information is retained at the leaves so that these trees can be recombined (Figure 2 (d)).

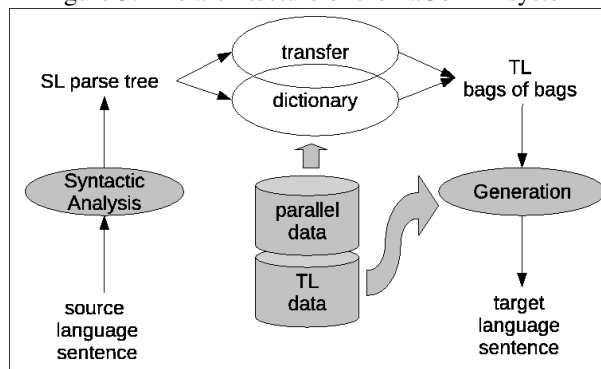
Figure 2 shows the four rules extracted from the alignments in Figure 2.3.. Rules are extracted by passing over the entire aligned treebank, identifying each aligned node pair and recursively iterating over its children to generate a substitutable pair of trees whose roots are aligned, and whose leaves are either terminal leaves in the treebank or correspond to aligned vertices. As shown in Figure 2, when a leaf node corresponds to an alignment point, we retain the information to identify which target tree leaf aligns with each such source leaf.

Many such tree substitution rules recur many times in the treebank, and a count is kept of the number of times each pair appears, resulting in a *stochastic* synchronous tree substitution grammar.

3.2. The PaCo-MT engine

Figure 3 presents the general system architecture of the PaCo-MT system (*Parse and Corpus-based MT*). A source language sentence gets syntactically analysed by a parser, that leads to a source language parse tree. We make an abstraction of the surface ordering in the source language parse tree and put the children of every node in a canonical order.

Figure 3: The architecture of the PaCo-MT system



We transfer this *unordered* parse tree by applying *tree transduction* with the *transfer grammar*, i.e. the grammar rules as extracted from the parallel treebank, as explained in section 3.1.. Note that the grammar also makes an abstraction of the surface ordering of both source and target sides.

The transducer takes an unordered source language parse tree and applies the STSG. For every node in the source parse tree, it looks for a grammar rule where the source lan-

guage side matches any horizontally complete subtree with the current node as its root.

When finding such a grammar rule, the corresponding target language side is inserted into the output forest.

The weight that the target side of the grammar rule will get is calculated according to Formula 1. We multiply the weight of the grammar rule with the relative frequency of the grammar rule over all grammar rules with the same source side. This is divided by an alignment point penalty, favouring the solutions with the least alignment points.

$$W = \frac{w(g)}{(j+1)^{app}} \times \frac{F(g)}{F(d^g)} \quad (1)$$

where

$w(g) = \sqrt[n]{\prod_{i=1}^n w(A_i^g)}$ is the weight of the grammar rule, which is the geometric mean of the weight of each individual occurrence of alignment A , as produced by `Lingua::Align`;

j is the number of alignment points, which is the number of non-lexical frontier elements that are aligned in the grammar rule;

app is the alignment points power parameter ($app = 0.5$);

$F(g)$ is the frequency of occurrence of the grammar rule g in the data;

$F(d^g)$ is the frequency of occurrence of the source side d of g in the data.

The output of the transducer is a set of target language bag of bags, i.e. a set of unordered trees, which we represent as a *weighted packed forest*. Figure 4 shows an example of a packed forest. More details about the transducer and the PaCo-MT engine can be found in (Vandeghinste et al., in press).

From this forest, the *tree-based target language generator* determines the surface order and generates an n -best list of translation alternatives by using a tree-based target language model. This is described in detail in (Vandeghinste, 2009).

3.3. Experiments and results

In this section, we compare the results of the transduction approach described in section 3.2. with the results from (Vandeghinste and Martens, 2009), in which we applied a top-down approach towards transduction and (Vandeghinste and Martens, 2010), in which we applied a bottom-up approach to transduction with on the fly rule generation.

We evaluated translation quality from Dutch to English using BLEU (Papineni et al., 2002), NIST (Doddington, 2002), and translation edit rate (TER) (Snover et al., 2006). In table 5 we compare the evaluation results of this approach with the results from (Vandeghinste and Martens, 2009) and from (Vandeghinste and Martens, 2010). There is a 7.9% relative rise of BLEU score between the approach from 2010 with the current approach. When we use the same data, but with alignments focussing on a high recall instead of a high precision, the BLEU score goes up with another 14.36% relative rise. While we have made a large progress in translation quality, we do not yet reach the same quality level as Moses (Koehn et al., 2007), which reaches a stunning BLEU score of 41.74 trained on Europarl alone.

Table 6 shows the effect of adding data to our model, gradually adding the subcorpora to the training data. This shows consistently better results for the conditions trained on data where alignments were determined focussing on recall rather than precision. The effect of adding data to the model is not entirely as expected. Overall, we can say that the quality improves when adding more data, but this is not the case when adding DGT-TM. The best results up till now have been reached in the condition where we omit the data from DGT-TM. A detailed analysis of the results will have to reveal the reasons for this.

4. Conclusion and future work

We have described our work on the creation of a large collection of richly annotated parallel treebanks that are aligned on multiple levels. Furthermore, we have applied these corpora to the development of a syntax-based MT system using STSG grammars. Although we are still making progress, there is still room for improvement, and we have not reached the same quality level as Moses.

As all parts of the MT engine, apart from the monolingual parsers, are language independent, we expect the approach to be transposable to other language pairs. In future work we will investigate how well this approach does on Dutch-French, English-Dutch and French-Dutch.

Current results suggest that for our data setup, high recall alignments are preferable to high precision alignments in producing better MT results. We achieve this by combining statistical with rule-based alignment methods that assume reasonably accurate word alignments and a degree of isomorphism between the trees produced by the source and target side parsers. For Dutch to English, at least, it has proven to work reasonably well.

There is much that can be done to increase alignment accuracy. Of course, more training data and further tuning of parameters are likely to lead to better scores. It is also possible to specify additional features to be integrated into `Lingua-Align`. We are currently also investigating the effect of more automatic error-correcting measures in the form of a transformation-based learning system. Finally, the impact of using different parsers and word alignment software still has to be investigated more thoroughly.

5. Acknowledgement

The work described in this paper was carried out within the PaCo-MT project. The PaCo-MT project was carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments (<http://www.stevintst.org>).

6. References

- C. Boitet and M. Tomokiyo. 1995. Ambiguities and ambiguity labelling: towards ambiguity data bases. In R. Mitkov and N. Nicolov, editors, *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, Tsigov Chark, Bulgaria.
- D. Chiang. 2006. An introduction to synchronous grammars. ACL Tutorial.

- B. de Marneffe, M. and MacCartney and C. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th edition of the International Conference on Language Resources and Evaluation (LREC)*.
- G. Doddington. 2002. Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. In *Proceedings of the Human Language Technology Conference (HLT)*, pages 128–132, San Diego, USA.
- J. Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proceedings of the 41st Annual Meeting of the ACL*, pages 205–208. ACL.
- D. Groves, M. Hearne, and A. Way. 2004. Robust sub-sentential alignment of phrase-structure trees. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-2004)*, pages 1072–1078.
- P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, Dyer D., O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 177–180, Prague, Czech Republic.
- P. Koehn. 2005. Europarl: a parallel corpus for statistical machine translation. In *Proceedings of MT Summit X*, pages 79–97. IAMT.
- G. Kotz . 2011a. Improving syntactic tree alignment of aligned parallel treebanks. In *Proceedings of ESSLLI 2011 Student Session*, pages 122–127, Ljubljana, Slovenia.
- G. Kotz . 2011b. Rule-induced error correction of aligned parallel treebanks. In *Proceedings of Corpus Linguistics*, pages 35–40, Saint Petersburg, Russia.
- A. Lavie, A. Parlikar, and V. Ambati. 2008. Syntax-driven learning of sub-sentential translation equivalents and translation rules from parsed parallel corpora. In *Proc. SSST*, pages 87–95.
- J. Lundborg, T. Marek, M. Mettler, and M. Volk. 2007. Using the stockholm treealigner. In *Proceedings of the 6th Workshop on Treebanks and Linguistic Theories*, pages 73–78, Bergen, Norway.
- A. Menezes and S.D. Richardson. 2001. A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In *Proceedings of the ACL 2001 Workshop on Data-Driven Methods in Machine Translation*, pages 39–46, Toulouse, France.
- R. Nesson, S. Shieber, and A. 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*, pages 128–137, Cambridge, USA.
- F.J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Y. Schabes. 1990. *Mathematical and Computational Aspects of Lexicalized Grammars*. Ph.D. thesis, University of Pennsylvania,.
- S. Shieber and Y. Schabes. 1990. Synchronous tree-adjointing grammars. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING)*, volume 3, pages 1–6.
- M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*.
- J. Tiedemann and G. Kotz . 2009. A discriminative approach to tree alignment. In *Proceedings of Recent Advances in Natural Language Processing (RANLP-2009)*.
- J. Tiedemann. 2009. News from opus - a collection of multilingual parallel corpora with tools and interfaces. In *Proceedings of Recent Advances in Natural Language Processing (RANLP-2009)*, pages 237–248.
- J. Tiedemann. 2010. Lingua-align: An experimental toolbox for automatic tree-to-tree alignment. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'2010)*, Valetta, Malta.
- G.J. van Noord. 2006. At last parsing is now operational. In *Proceedings of Traitement Automatique des Langues Naturelles (TALN)*, pages 20–42.
- V. Vandeghinste and S. Martens. 2009. Top-down transfer in example-based mt. In *Proceedings of the 3rd Workshop on Example-based Machine Translation*, pages 69–76, Dublin, Ireland.
- V. Vandeghinste and S. Martens. 2010. Bottom-up transfer in example-based machine translation. In *Proceedings of the 14th International Conference of the European Association for Machine Translation (EAMT-2010)*, Saint-Rapha l, France.
- V. Vandeghinste, S. Martens, G. Kotz , J. Tiedemann, J. Van den Bogaert, K. De Smet, F. Van Eynde, and G. van Noord. in press. Parse and corpus-based machine translation. In *Essential Speech and Language Technology for Dutch: resources, tools and applications*. Springer.
- V. Vandeghinste. 2009. Tree-based target language modeling. In *Proceedings of the 13th International Conference of the European Association for Machine Translation (EAMT-2009)*, Barcelona, Spain.
- D. Varga, L. N meth, P. Hal csy, A. Kornai, V. Tr n, and V. Nagy. 2005. Parallel corpora for medium density languages. In *Proceedings of Recent Advances in Natural Language Processing (RANLP-2005)*, pages 590–596.
- V. Zhechev and A. Way. 2008. Automatic generation of parallel treebanks. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING-2008)*, pages 1105–1112.

Figure 4: An example of a packed forest as output of the transducer for the Dutch sentence *Het heeft ook een wettelijke reden*. Note that ? marks an alternation.

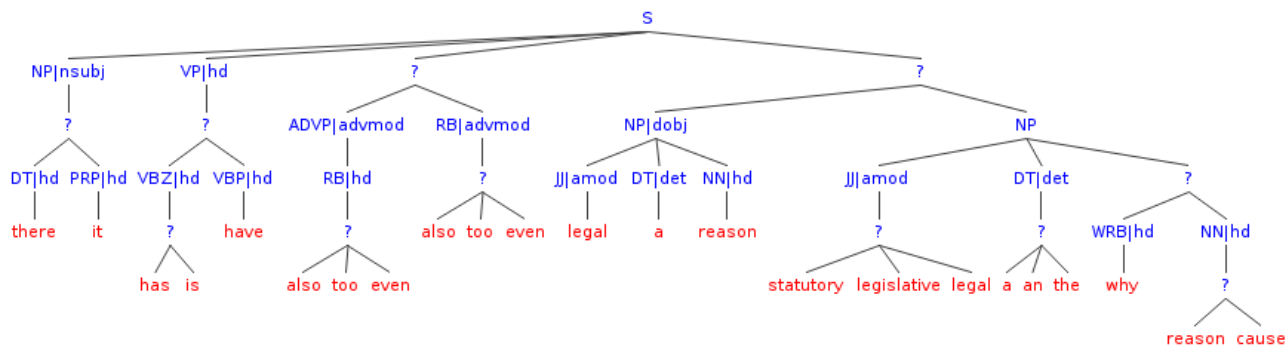


Table 5: Comparing the different transduction approaches for Dutch to English trained on Europarl

Transducer	Prec/Recall	BLEU	NIST	TER
V&M (2009)	Precision	13.53	5.70	70.36
V&M (2010)	Precision	20.65	6.44	63.72
PaCo-MT	Precision	22.28	6.43	64.14
PaCo-MT	Recall	25.48	7.36	61.12

Table 6: Evaluation of adding data to the model

Treebanks	High Precision			High Recall		
	BLEU	NIST	TER	BLEU	NIST	TER
Europarl	22.28	6.43	64.14	25.48	7.36	61.12
Europarl+OPUS	22.71	6.57	64.68	26.23	7.40	61.63
Europarl+OPUS+DGT-TM	22.83	6.66	65.26	24.10	6.59	64.08
Europarl+OPUS+DGT-TM+Transmem	24.19	6.78	64.31	26.57	6.95	62.46
Europarl+OPUS+Transmem				29.12	7.68	60.04