

Chapter 17

Parse and Corpus-Based Machine Translation

Vincent Vandeghinste, Scott Martens, Gideon Kotzé, Jörg Tiedemann,
Joachim Van den Bogaert, Koen De Smet, Frank Van Eynde,
and Gertjan van Noord

17.1 Introduction

The current state-of-the-art in machine translation consists of *phrase-based statistical machine translation (PB-SMT)* [23], an approach which has been used since the late 1990s, evolving from word-based SMT proposed by IBM [5]. These *string-based* techniques (which use no linguistic knowledge) seem to have reached their ceiling in terms of translation quality, while there are still a number of limitations to the model. It lacks a mechanism to deal with long-distance dependencies, it has no means to generalise over non-overt linguistic information [37] and it has limited word reordering capabilities. Furthermore, in some cases the output quality may lack appropriate fluency and grammaticality to be acceptable for actual MT users. Sometimes essential words are missing from the translation.

To overcome these limitations efforts have been made to introduce *syntactic knowledge* into the statistical paradigm, usually in the form of syntax trees, either

V. Vandeghinste (✉) · J. Van den Bogaert · F. Van Eynde
Centrum voor Computerlinguïstiek (CCL), Leuven University, Leuven, Belgium
e-mail: vincent@ccl.kuleuven.be; joachim@ccl.kuleuven.be; frank.vaneynde@ccl.kuleuven.be

S. Martens
University of Tübingen (previously at CCL), Tübingen, Germany
e-mail: scott.martens@uni-tuebingen.de

G. Kotzé · G. van Noord
Groningen University, Groningen, The Netherlands
e-mail: g.j.kotze@rug.nl; g.j.m.van.noord@rug.nl

J. Tiedemann
University of Uppsala (previously at Groningen University), Uppsala, Sweden
e-mail: jorg.tiedemann@lingfil.uu.se

K. De Smet
Oneliner bvba, Sint-Niklaas, Belgium
e-mail: koen@oneliner.be

only for the source (tree-to-string) or the target language (string-to-tree), or for both (tree-to-tree).

Galley et al. [12] describes an MT engine in which *tree-to-string* rules have been derived from a parallel corpus, driven by the problems of SMT systems raised by [11]. Marcu et al. and Wang et al. [30, 52] describe *string-to-tree* systems to allow for better reordering than phrase-based SMT and to improve grammaticality. Hassan et al. [18] implements another string-to-tree system by means of including supertags [2] to the target side of the phrase-based SMT baseline.

Most of the *tree-to-tree* approaches use one or another form of *synchronous context-free grammars* (SCFGs) a.k.a. syntax directed translations [1] or syntax directed *transduction* grammars [28]. This is true for the tree-based models of the Moses toolkit,¹ and the machine translation techniques described in, amongst others [7, 27, 36, 53–55]. A more complex type of translation grammars is *synchronous tree substitution grammar* (STSG) [10, 38] which provides a way, as [8] points out, to perform certain operations which are not possible with SCFGs without flattening the trees, such as raising and lowering nodes. Examples of STSG approaches are the *Data-Oriented Translation* (DOT) model from [20, 35] which uses data-oriented parsing [3] and the approaches described in [14–16] and [37], using STSG rules consisting of dependency subtrees, and a top-down transduction model using beam search.

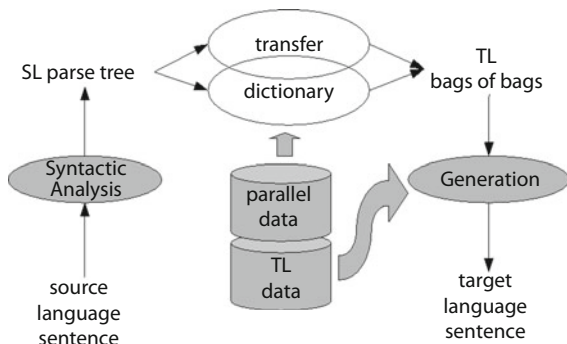
The *Parse and Corpus based MT* (PaCo-MT) engine described in this chapter² is another tree-to-tree system that uses an STSG, differing from related work with STSGs in that the PaCo-MT engine combines dependency information with constituency information and that the translation model abstracts over word and phrase order in the synchronous grammar rules: the daughters of any node are in a canonical order representing all permutations. The final word order is generated by the tree-based target language modeling component.

Figure 17.1 presents the architecture of the PaCo-MT system. A source language (SL) sentence gets syntactically analysed by a pre-existing parser which leads to a source language parse tree, making abstraction of the surface order. This is described in Sect. 17.2. The unordered parse tree is translated into a forest of unordered trees (a.k.a. bag of bags) by applying *tree transduction* with the *transfer grammar* which is an STSG derived from a parallel treebank. Section 17.3 presents how the transduction grammar was built and Sect. 17.4 how this grammar is used in the translation process. The forest is decoded by the *target language generator*, described in Sect. 17.5 which generates an *n*-best list of translation alternatives by using a tree-based target language model. The system is evaluated on Dutch to English in Sect. 17.6 and conclusions are drawn in Sect. 17.7. As all modules of our system are language independent results for Dutch → French, English → Dutch, and French → Dutch can be expected soon.

¹<http://www.statmt.org/moses/>

²Previous versions were described in [48] and [49].

Fig. 17.1 The architecture of the PaCo-MT system



17.2 Syntactic Analysis

Dutch input sentences are parsed using Alpino [32], a stochastic rule-based dependency parser, resulting in structures as in Fig. 17.2.³

In order to induce the translation grammar, as explained in Sect. 17.3, parse trees for the English sentences in the parallel corpora are also required. These sentences are parsed using the Stanford phrase structure parser [21] with dependency information [31]. The bracketed phrase structure and the typed dependency information are integrated into an XML format consistent with the Alpino XML format. All tokens are lemmatised using TreeTagger [39].

Abstraction is made of the surface order of the terminals in every parse tree used in the PaCo-MT system. An *unordered tree* is defined⁴ by the tuple $\langle V, V^i, E, L \rangle$ where V is the set of nodes, V^i is the set of internal nodes, and $V^f = V - V^i$ is the set of frontier nodes, i.e. nodes without daughters. $E \subset V^i \times V$ is the set of directed edges and L is the set of labels on nodes or edges. $V^l \subseteq V^f$ is the set of lexical frontier nodes, containing actual words as labels, and $V^n = V^f - V^l$ is the set of non-lexical frontier nodes, which is empty in a full parse tree, but not necessarily in a subtree. There is exactly one root node $r \in V^i$ without incoming edges. Let T be the set of all unordered trees, including subtrees.

A subtree $s_r \in T$ of a tree $t \in T$ has as a root node $r \in V_t^i$ where V_t^i is the set of internal nodes of t . Subtrees are *horizontally complete* [4] if, when a daughter node of a node is included in the subtree, then so are all of its sisters. Figure 17.3 shows an example. Let $H \subset T$ be the set of all horizontally complete subtrees.

Bottom-up subtrees are a subset of the horizontally complete subtrees: they are lexical subtrees: every terminal node of the subtree is a lexical node. Some examples are shown in Fig. 17.4. Let $B \subset H$ be the set of all bottom-up subtrees. $\forall b \in B$:

³Limited restructuring is applied to make the resulting parse trees more uniform. For instance, nouns are always placed under an NP. A similar restructuring of syntax trees is shown by [52] to improve translation results.

⁴This definition is inspired by [10].

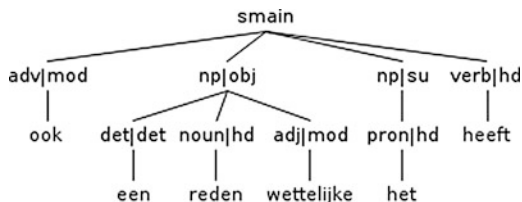


Fig. 17.2 An unordered parse tree for the Dutch sentence *Het heeft ook een wettelijke reden* “It also has a legal reason”, or according to Europarl “It is also subject to a legal requirement”. Note that edge labels are marked behind the ‘|’

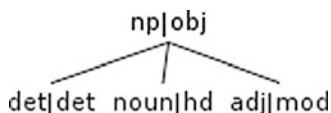


Fig. 17.3 An example of a horizontally complete subtree which is not a bottom-up subtree

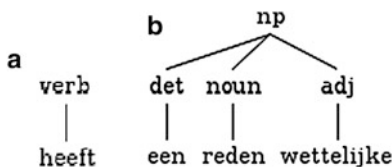


Fig. 17.4 Two examples of bottom-up subtrees

$V_b^n = \emptyset$ and $V_b^l = V_b^f$, where V_b^n is the set of non-lexical frontier nodes of b and V_b^l is the set of lexical frontier nodes of b . V_b^f is the set of all frontier nodes of b .

17.3 The Transduction Grammar

In order to translate a source sentence, a stochastic synchronous tree substitution grammar G is applied to the source sentence parse tree. Every grammar rule $g \in G$ consists of an elementary tree pair, defined by the tuple $\langle d^g, e^g, A^g \rangle$, where $d^g \in T$ is the source side tree (Dutch), $e^g \in T$ is the target side tree (English), and A^g is the alignment between the non-lexical frontier nodes of d^g and e^g . The alignment A^g is defined by a set of tuples $\langle v_d, v_e \rangle$ where $v_d \in V_d^n$ and $v_e \in V_e^n$. V_d^n is the set of non-lexical frontier nodes of d^g , and V_e^n is the set of non-lexical frontier nodes of e^g . Every non-lexical frontier node of the source side is aligned with a non-lexical frontier node of the target side: $\forall v_d \in V_d^n$ is aligned with a node $v_e \in V_e^n$. An example grammar rule is shown in Fig. 17.5.

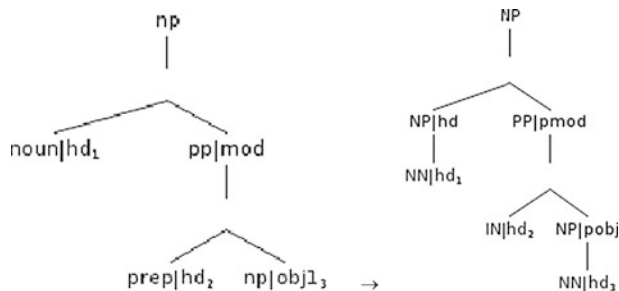


Fig. 17.5 An example of a grammar rule with horizontally complete subtrees on both the source and target side. Indices mark alignments

In order to induce such a grammar a *node aligned* parallel treebank is required. Section 17.3.1 describes how to build such a treebank. Section 17.3.2 describes the actual induction process.

17.3.1 Preprocessing and Alignment of the Parallel Data

The system was trained on the Dutch-English subsets of the Europarl corpus [22], the DGT translation memory,⁵ the OPUS corpus⁶ [42] and an additional private translation memory (transmem).

The data was syntactically parsed (as described in Sect. 17.2), sentence aligned using Hunalign [50] and word aligned using GIZA++ [33]. The bidirectional GIZA++ word alignments were refined using the *intersect* and *grow-diag* heuristics implemented by Moses [24], resulting in a higher recall for alignments suitable for machine translation.

For training Lingua-Align [43], which is a discriminative tree aligner [44], a set of parallel alignments was manually constructed using the Stockholm TreeAligner [29], for which the already existing word alignments were imported. The recall of the resulting alignments was rather low, even though in constructing the training data a more relaxed version of the well-formedness criteria as proposed by [19] was used.

Various features and parameters have been used in experimentation, training with around 90 % and testing with the rest of the data set. The training data set consists of 140 parallel sentences.

Recent studies in rule-based alignment error correction ([25, 26]) show that recall can be significantly increased while retaining a relatively high degree of precision.

⁵<http://langtech.jrc.it/DGT-TM.html>

⁶<http://opus.lingfil.uu.se/>

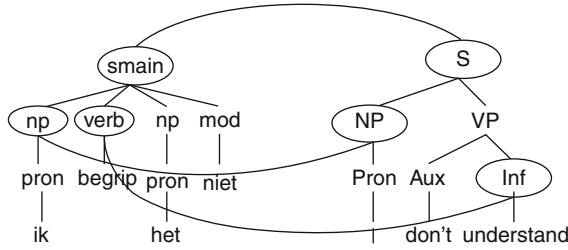


Fig. 17.6 Two sentences with subsentential alignment

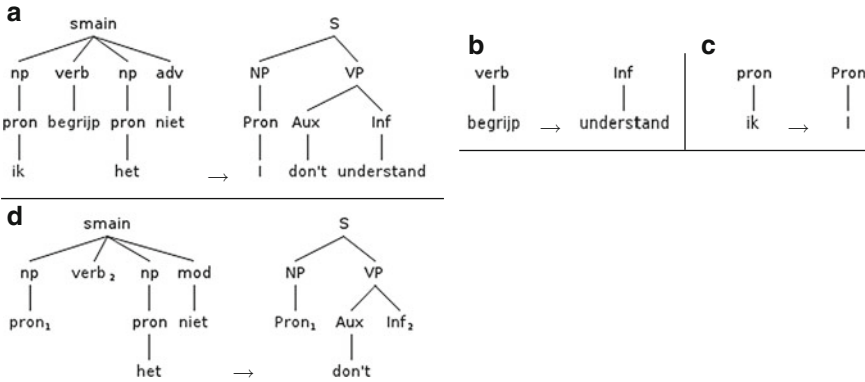


Fig. 17.7 Rules extracted from the alignments in Fig. 17.6

This approach has been extended by applying a bottom-up rule addition component that greedily adds alignments based on already existing word alignments, more relaxed well-formedness criteria, as well as using measures of similarities between the two unlinked subtrees being considered for alignment.

17.3.2 Grammar Rule Induction

Figure 17.6 is an example⁷ of two sentences aligned at both the sentence and subsentential 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 (Fig. 17.7a, b).

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

⁷The edge labels have been omitted from these examples, but were used in the actual rule induction.

rules, the alignment information is retained at the leaves so that these trees can be recombined (Fig. 17.7d).

The rule extraction process was restricted to rules with horizontally complete subtrees at the source and target side. Rule extraction with other types of subtrees was considered out of the scope of the current research.

Figure 17.7 shows the four rules extracted from the alignments in Fig. 17.6. 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 Fig. 17.7, 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.

17.4 The Transduction Process

The *transduction process* takes an unordered source language parse tree $p \in T$ as input, applies the transduction grammar G and transduces p into an unordered weighted packed forest, which is a compact representation of a set of target trees $Q \subset T$, which represent the translation alternatives. An example of a packed forest is shown in Fig. 17.8.

For every node $v \in V_p^i$, where V_p^i is the set of internal nodes in the input parse tree p , it is checked whether there is a subtree $s_v \in H$ with v as its root node, which matches the source side tree d^g of a grammar rule $g \in G$.

To keep computational complexity limited the subtrees of p that are considered and the subtrees that occur in the source and target side of the grammar G have been restricted to horizontally complete subtrees (including bottom-up subtrees).

When finding a matching grammar rule for which $s_v = d^g$, the corresponding e^g is inserted into the output forest Q . When not finding a matching grammar rule, a horizontally complete subtree is constructed, as explained in Sect. 17.4.2.

The weight that the target side e^g of grammar rule $g \in G$ will get when is calculated according to Eq. 17.1. This weight calculation is similar to the approaches of [14, 37], as it contains largely the same factors. We multiply the weight of the grammar rule $w(g)$ with the relative frequency of the grammar rule over all grammar rules with the same source side $\frac{F(g)}{F(d^g)}$. This is divided by an alignment point penalty $(j + 1)^{app}$, favouring the solutions with the least alignment points.

$$W(e^g) = \frac{w(g)}{(j + 1)^{app}} \times \frac{F(g)}{F(d^g)} \quad (17.1)$$

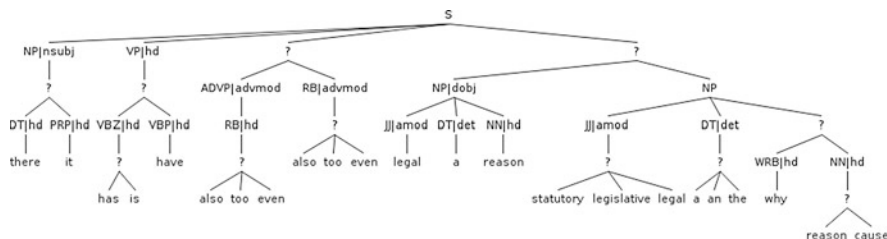


Fig. 17.8 An example of a packed forest as output of the transducer for the Dutch sentence *Het heeft ook een wettelijke reden.* Note that ? marks an alternation

where $w(g) = \sqrt[n]{\prod_{i=1}^n w(A_i^g)}$ is the weight of $g \in G$, which is the geometric mean of the weight of each individual occurrence of alignment A , as produced by the discriminative aligner described in Sect. 17.3.1; $j = |V_d^n| = |V_e^n|$ is the number of alignment points, which is the number of non-lexical frontier elements which are aligned in $g \in G$; app is the alignment points power parameter ($app = 0.5$); $F(g)$ is the frequency of occurrence g in the data; $F(d^g)$ is the frequency of occurrence of the source side d of g in the data.

When no translation of a word is found in the transduction grammar, the label $l \in L$ is mapped onto its target language equivalent. Adding a simple bilingual word form dictionary is optional. When a word translation is not found in the transduction grammar, the word is looked up in this dictionary. If the word has multiple translations in the dictionary, each of these translations receives the same weight and is combined with the translated label (usually part-of-speech tags). When the word is not in the dictionary or no dictionary is present, the source word is transferred as is to Q .

17.4.1 Subtree Matching

In a first step, the transducer performs *bottom-up subtree* matching, which is analogous to the use of phrases in phrase-based SMT, but restricted to linguistically meaningful phrases. Bottom-up subtree matching functions like a sub-sentential translation memory: every linguistically meaningful phrase that has been encountered in the data will be considered in the transduction process, obliterating the distinction between a translation memory, a dictionary and a parallel corpus [45].

For every node $v \in V_p$ it is checked whether a subtree s_v with root node v is found for which $s_v \in B$ and for which there is a grammar rule $g \in G$ for which $d = s_v$. These matches include single word translations together with their parts-of-speech.

A second step consists of performing *horizontally complete subtree* matching for those nodes in the source parse tree for which the number of grammar rules $g \in G$ that match is smaller than the beam size b .

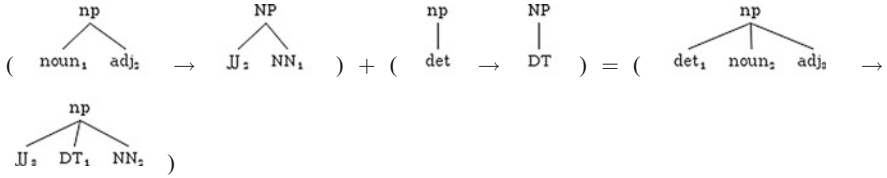


Fig. 17.9 An example of a constructed grammar rule

For every node $v \in V_p^i$ the set $H_v \subset H \setminus B$ is generated, which is the set of all horizontally complete subtrees minus the bottom-up subtrees of p with root node v . It is checked whether a matching subtree $s_v \in H_v$ is found for which there is a grammar rule $g \in G$ for which $d^g = s_v$.

An example of a grammar rule with horizontally complete subtrees on both source and target sides was shown in Fig. 17.5. This rule has three alignment points, as indicated by the indices.

17.4.2 Backing Off to Constructed Horizontally Complete Subtrees

In cases where no grammar rules are found for which the source side matches the horizontally complete subtrees at a certain node in the input parse tree, grammar rules are combined for which, when combined, the source sides form a horizontally complete subtree. An example of such a constructed grammar rule is shown in Fig. 17.9.

$\forall v \in V_p^i$ for which there is no $s_v \in H_v$ matching any grammar rule $g \in G$, let $C_s = \langle c_1, \dots, c_n \rangle$ be the set of children of root node v in subtree s_v . $\forall c_j \in C_s$ the subtree s_v is split into two partial subtrees y_v and z_v , where $C_y = C_s \setminus \{c_j\}$ is the set of children of subtree y_v and $C_z = \{c_j\}$ is the set of children of subtree z_v .

When a grammar rule $g \in G$ is found for which $d^g = y_v$ and another grammar rule $h \in G$ is found for which $d^h = z_v$, then the respective target sides e_q^g with root node q and e_u^h with root node u are merged into one target language tree e^f if $q = u$ and $C_{e^g, h} = C_{e^g} \cup C_{e^h}$, resulting in a constructed grammar rule $f \notin G$ defined by the tuple $\langle d^f, e^f, A^f \rangle$, where $d^f = s_v$. The alignment of the constructed grammar rule is the union of the alignments of the grammar rules g and h : $A^f = A^g \cup A^h$.

As f is a constructed grammar rule, the absolute frequency of occurrence of the grammar rule $F(f) = 0$, which would result in $W(e^{g,h}) = 0$ in Eq. 17.1. In order to resolve this, the frequency of occurrence $F(f)$ is estimated according to Eq. 17.2.

$$F(f) = w(y_v) \times \frac{F(g)}{F(d^g)} \times \frac{F(h)}{F(d^h)} \tag{17.2}$$

where

- $w(y_v) = \sqrt[m]{\prod_{i=1}^m w(A_i^g)}$ is the weight of grammar rule g , which is the geometric mean of the weight of each individual occurrence of alignment A , as produced by the discriminative aligner described in 17.3.1;
- $F(g)$ is the frequency of occurrence of grammar rule g
- $F(d^g)$ is the frequency of occurrence of the source side d^g of grammar rule g
- $F(h)$ is the frequency of occurrence of grammar rule h
- $F(d^h)$ is the frequency of occurrence of the source side d^h of grammar rule g^h

Constructing grammar rules leads to overgeneration. As a filter the target language probability of such a rule is taken into account. This is estimated by multiplying the relative frequency of v_j in which c_i occurs as a child over all v_j 's with the relative frequency of c_j occurring N times over c_j occurring any number of times, as shown in Eq. 17.3, which is applied recursively for every node $v_j \in V_e$ where V_e is the set of nodes in e^f .

$$P(e^f) = \prod_{j=1}^m \prod_{i=1}^n \frac{F(\#(c_i|v_j) \geq 1)}{F(v_j)} \times \frac{F(\#(c_i|v_j) = N)}{\sum_{r=1}^n F(\#(c_i|v_j) = r)} \quad (17.3)$$

where

$\#(c_i|v_j)$ is the number of children of v_j with the same label as c_i

N is the number of times the label c_i occurs in the constructed rule

The new weight $w(e^f)$ is calculated according to Eq. 17.4.

$$w(e^f) = \sqrt[cp]{F(f) \times P(e^f)} \quad (17.4)$$

where

cp is the construction penalty: $0 \leq cp \leq 1$.

When constructing a horizontally complete subtree fails, a grammar rule is constructed by translating each child separately.

17.5 Generation

The main task of the *target language generator* is to determine word order, as the packed forest contains unordered trees. An additional task of the target language model is to provide additional information concerning lexical selection, similar to the language model in phrase-based SMT [23].

The target language generator has been described in detail in [47], but the system has been generalised and improved and was adapted to work with weighted packed forests as input.

For every node in the forest, the surface order of its children needs to be determined. For instance, when translating “*een wettelijke reden*” into English, the bag $NP\langle JJ(\textit{legal}), DT(a), NN(\textit{reason}) \rangle$ represents the surface order of all permutations of these elements.

A large monolingual treebank is searched for an NP with an occurrence of these three elements, and in what order they occur most, using the relative frequency of each permutation as a weight. If none of the permutations are found, the system backs off to a more abstract level, only looking for the bag $NP\langle JJ, DT, NN \rangle$ without lexical information, for which there is most likely a match in the treebank.

When still not finding a match, all permutations are generated with an equal weight, and a penalty is applied for the distance between the source language word order and the target language word order to avoid generating too many solutions with exactly the same weight. This is related to the notion of distortion in IBM model 3 in [5].

In the example bag, there are two types of information for each child: the part-of-speech and the word token, but as already pointed out in Sect. 17.2 dependency information and lemmas are also at our disposal.

All different information sources (token, lemma, part-of-speech, and dependency relation) have been investigated with a back-off from most concrete (token + lemma + part-of-speech + dependency relation) to most abstract (part-of-speech).

The functionality of the generator is similar to the one described in [17], but relative frequency of occurrence is used instead of n -grams of dependencies. As shown in [47] this approach outperforms SRILM 3-g models [41] for word ordering. [51] uses *feature templates* for translation candidate reranking, but these can have a higher depth and complexity than the context-free rules used here.

Large monolingual target language treebanks have been built by using the target sides of the parallel corpora and adding the British National Corpus (BNC)⁸.

17.6 Evaluation

We evaluated translation quality from Dutch to English on a test set of 500 sentences with three reference translations, using BLEU [34], NIST [9] and translation edit rate (TER) [40], as shown in Table 17.1.

We show the effect of adding data, by presenting the results when using the Europarl (EP) corpus, and when adding the OPUS corpus, the DGT corpus, and the private translation memory (transmem), and we show the effect of adding a dictionary of + 100,000 words, taken from the METIS Dutch English translation engine [6, 46]. This dictionary is only used for words where the grammar does not cover a translation.

⁸<http://www.natcorp.ox.ac.uk/>

Table 17.1 Evaluation of the Dutch-English engine

Training data	Without dictionary			With dictionary		
	BLEU	NIST	TER	BLEU	NIST	TER
EP	25.48	7.36	61.12	25.75	7.43	60.38
EP+OPUS	26.23	7.40	61.63	26.46	7.44	61.42
EP+OPUS+DGT	24.10	6.59	64.08	25.82	7.28	61.83
EP+OPUS+transmem	29.12	7.68	60.04	29.33	7.71	59.98
EP+OPUS+DGT+transmem	28.50	7.59	60.22	29.31	7.71	59.47

These results show that the best scoring condition is trained on all the data apart from DGT, which seems to deteriorate performance. Adding the dictionary is beneficial under all conditions. Error analysis shows that the system often fails when using the back-off models, whereas it seems to function properly when horizontally complete subtrees are found.

Comparing the results with Moses⁹ [24] shows that there is a long way to go for our syntax-based approach until we par with phrase-based SMT. The difference in score is partly due to remaining bugs in the PaCo-MT system which cause no output in 2.6 % of the cases. Another reason could be the fact that automated metrics like BLEU are known to favour phrase-based SMT systems. Nevertheless, the PaCo-MT system has not yet reached its full maturity and there are several ways to improve the approach, as discussed in Sect. 17.7.

17.7 Conclusions and Future Work

With the research presented in this paper we wanted to investigate an alternative approach towards MT, not using n -grams or any other techniques from phrase-based SMT systems.¹⁰

A detailed error analysis and comparison between the different conditions will reveal what can be done to improve the system. Different parameters in alignment can result in more useful information from the same set of data. Different approaches to grammar induction could also improve the system, as grammar induction is now limited to horizontally complete subtrees. STSGs allow more complex grammar rules including horizontally incomplete subtrees. Another improvement can be expected from working on the back-off strategy in the transducer, such as the real time construction of new grammar rules on the basis of partial grammar rules.

⁹This phrase-based SMT system was trained on the same test set with the same training data, using 5-g without minimum error rate training scored 41.74, 43.30, 44.46, 49.61 and 49.98 BLEU respectively.

¹⁰Apart from word alignment.

The system could be converted into a syntactic translation aid, by only taking the decisions of which it is confident, backing off to human decisions in cases of data sparsity. It remains to be tested whether this approach would be useful.

Further investigation of the induced grammar could lead to a reduction in grammar rules, by implementing a default inheritance hierarchy, similar to [13], speeding up the system, without having any negative effects on the output.

The current results of our system are in our opinion not sufficient to reject nor accept a syntax-based approach towards MT as an alternative for phrase-based SMT, as, quoting Kevin Knight “*the devil is in the details*”.¹¹

Open Access. This chapter is distributed under the terms of the Creative Commons Attribution Noncommercial License, which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Aho, A., Ullman, J.: Syntax directed translations and the pushdown assembler. *J. Comput. Syst. Sci.* **3**, 37–56 (1969)
2. Bangalore, S., Joshi, A. (eds.): *Supertagging*. MIT, Cambridge, Massachusetts (2010)
3. Bod, R.: A Computational Model of Language Performance: Data-Oriented Parsing. In: *Proceedings of the 15th International Conference on Computational Linguistics (COLING)*, Nantes, France, pp. 855–856 (1992)
4. Boitet, C., Tomokiyo, M.: Ambiguities and ambiguity labelling: towards ambiguity data bases. In: R. Mitkov, N. Nicolov (eds.) *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, Tsigov Chark, Bulgaria (1995)
5. Brown, P., Cocke, F., Della Pietra, S., V.J., D.P., Jelinek, F., Lafferty, J., Mercer, R., Roossin, P.: A statistical approach to machine translation. *Comput. Linguist.* **16**(2), 79–85 (1990)
6. Carl, M., Melero, M., Badia, T., Vandeghinste, V., Dirix, P., Schuurman, I., Markantonatou, S., Sofianopoulos, S., Vassiliou, M., Yannoutsou, O.: METIS-II: low resources machine translation : background, implementation, results, and potentials. *Mach. Trans.* **22**(1), 67–99 (2008)
7. Chiang, D.: A hierarchical phrase-based model for statistical machine translation. In: *Proceedings of the 43rd Annual Meeting of the ACL*, Ann Arbor, US, pp. 263–270. *ACL* (2005)
8. Chiang, D.: An introduction to synchronous grammars. *COLING/ACL Tutorial*, Sydney, Australia (2006)
9. Doddington, G.: Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. In: *Proceedings of the Human Language Technology Conference (HLT)*, San Diego, USA, pp. 128–132 (2002)
10. Eisner, J.: Learning non-isomorphic tree mappings for machine translation. In: *Proceedings of the 41st Annual Meeting of the ACL*, Sapporo, Japan, pp. 205–208. *ACL* (2003)
11. Fox, H.: Phrasal cohesion and statistical machine translation. In: *Proceedings of the 2002 conference on Empirical Methods in Natural Language Processing*, Philadelphia, USA, pp. 304–311 (2002)
12. Galley, M., Hopkins, M., Knight, K., Marcu, D.: What’s in a translation rule? In: *Proceedings of the HLT Conference of the North American Chapter of the ACL (NAACL)*, Boston, USA, pp. 273–280 (2004)

¹¹Comment of Kevin Knight on the question why syntax-based MT does not consistently perform better or worse than phrase-based SMT, at the 2012 workshop “More Structure for Better Statistical Machine Translation?” held in Amsterdam.

13. Gazdar, G., Klein, E., Pullum, G., Sag, I.: *Generalized Phrase Structure Grammar*. Blackwell, Oxford, UK (1985)
14. Graham, Y.: Sulis: An Open Source Transfer Decoder for Deep Syntactic Statistical Machine Translation. *Prague Bull. Math. Linguist.* **93**, 17–26 (2010)
15. Graham, Y., van Genabith, J.: Deep Syntax Language Models and Statistical Machine Translation. In: *Proceedings of the 4th Workshop on Syntax and Structure in Statistical Translation (SSST-4)*, Beijing, China, pp. 118–126 (2010)
16. Graham, Y., van Genabith, J.: Factor templates for factored machine translation models. In: *Proceedings of the 7th International Workshop on Spoken Language Translation (IWSLT)*, Paris, France (2010)
17. Guo, Y., van Genabith, J., Wang, H.: Dependency-based N-gram Models for General Purpose Sentence Realisation. In: *Proceedings of the 22nd International Conference on Computational Linguistics (COLING)*, Manchester, UK, pp. 297–304 (2008)
18. Hassan, H., Sima'an, K., Way, A.: Supertagged phrase-based statistical machine translation. In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, pp. 288–295 (2007)
19. Hearne, M., Tinsley, J., Zhechev, V., Way, A.: 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)*, Skvde, Sweden (2007)
20. Hearne, M., Way, A.: Seeing the wood for the trees. *Data-Oriented Translation*. In: *Proceedings of MT Summit IX*, New Orleans, US (2003)
21. Klein, D., Manning, C.: Accurate unlexicalized parsing. In: *Proceedings of the 41st Annual Meeting of the ACL*, Sapporo, Japan, pp. 423–430. ACL (2003)
22. Koehn, P.: Europarl: a parallel corpus for statistical machine translation. In: *Proceedings of MT Summit X*, Phuket, Thailand, pp. 79–97. IAMT (2005)
23. Koehn, P.: *Statistical Machine Translation*. Cambridge (2010)
24. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., D., D., Bojar, O., Constantin, A., Herbst, E.: Moses: open source toolkit for statistical machine translation. In: *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, Prague, Czech Republic, pp. 177–180 (2007)
25. Kotzé, G.: Improving syntactic tree alignment through rule-based error correction. In: *Proceedings of ESSLLI 2011 Student Session*, Ljubljana, Slovenia, pp. 122–127 (2011)
26. Kotzé, G.: Rule-induced correction of aligned parallel treebanks. In: *Proceedings of Corpus Linguistics*, Saint Petersburg, Russia (2011)
27. Lavie, A.: Stat-xfer: A general serach-based syntax-driven framework for machine translation. In: *Proceedings of the 9th International Conference on Intelligent Text Processing and Computational Linguistics*, Haifa, Israel, pp. 362–375 (2008)
28. Lewis, P., Stearns, R.: Syntax-directed transduction. *J. ACM* **15**, 465–488 (1968)
29. Lundborg, J., Marek, T., Mettler, M., Volk, M.: Using the Stockholm TreeAligner. In: *Proceedings of the 6th Workshop on Treebanks and Linguistic Theories*, Bergen, Norway, pp. 73–78 (2007)
30. Marcu, D., Wang, W., Echihabi, A., Knight, K.: SPMT: statistical machine translation with syntactified target language phrases. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia (2006)
31. de Marneffe, M., MacCartney, B., Manning, C.: Generating typed dependency parses from phrase structure parses. In: *Proceedings of the 5th edition of the International Conference on Language Resources and Evaluation (LREC)*, Genoa, Italy (2006)
32. van Noord, G.: At last parsing is now operational. In: *Proceedings of Traitement Automatique des Langues Naturelles (TALN)*, Leuven, Belgium, pp. 20–42 (2006)
33. Och, F., Ney, H.: A systematic comparison of various statistical alignment models. *Comput. Linguist.* **29**(1), 19–51 (2003)
34. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318 (2002)

35. Poutsma, A.: Machine Translation with Tree-DOP. In: R. Bod, R. Scha, K. Sima'an (eds.) *Data-Oriented Parsing*, chap. 18, pp. 339–358. CSLI, Stanford, US (2003)
36. Probst, K., Levin, L., Peterson, E., Lavie, A., Carbonel, J.: MT for Minority Languages Using Elicitation-Based Learning of Syntactic Transfer Rules. *Mach. Trans.* **17**(4), 245–270 (2002)
37. Riezler, S., Maxwell III, J.: Grammatical Machine Translation. In: *Proceedings of the HLT Conference of the North American Chapter of the ACL (NAACL)*, New York, USA, pp. 248–255 (2006)
38. Schabes, Y.: *Mathematical and Computational Aspects of Lexicalized Grammars*. Ph.D. thesis, University of Pennsylvania, (1990)
39. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK (1994)
40. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: *Proceedings of Association for Machine Translation in the Americas* (2006)
41. Stolcke, A.: SRILM – an extensible language modeling toolkit. In: *Proceedings of the International Conference on Spoken Language Processing*, Denver, USA (2002)
42. Tiedemann, J.: News from OPUS – a collection of multilingual parallel corpora with Tools and Interfaces. In: *Proceedings of Recent Advances in Natural Language Processing (RANLP-2009)*, Borovets, Bulgaria, pp. 237–248 (2009)
43. Tiedemann, J.: 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 (2010)
44. Tiedemann, J., Kotzé, G.: A discriminative approach to tree alignment. In: *Proceedings of Recent Advances in Natural Language Processing (RANLP-2009)*, Borovets, Bulgaria (2009)
45. Vandeghinste, V.: Removing the distinction between a translation memory, a bilingual dictionary and a parallel corpus. In: *Proceedings of Translation and the Computer 29*, ASLIB, London, UK (2007)
46. Vandeghinste, V.: *A Hybrid Modular Machine Translation System. LoRe-MT: Low Resources Machine Translation*. Ph.D. thesis, K.U. Leuven, Leuven, Belgium (2008)
47. Vandeghinste, V.: Tree-based target language modeling. In: *Proceedings of the 13th International Conference of the European Association for Machine Translation (EAMT-2009)*, Barcelona, Spain (2009)
48. Vandeghinste, V., Martens, S.: Top-down transfer in example-based MT. In: *Proceedings of the 3rd Workshop on Example-based Machine Translation*, Dublin, Ireland, pp. 69–76 (2009)
49. Vandeghinste, V., Martens, S.: 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-Raphal, France (2010)
50. Varga, D., Németh, L., Halácsy, P., Kornai, A., Trón, V., Nagy, V.: Parallel corpora for medium density languages. In: *Proceedings of Recent Advances in Natural Language Processing (RANLP-2005)*, Borovets, Bulgaria, pp. 590–596 (2005)
51. Velldal, E., Oepen, S.: Statistical ranking in tactical generation. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia (2006)
52. Wang, W., May, J., Knight, K., Marcu, D.: Re-structuring, re-labeling, and re-aligning for syntax-based machine translation. *Comput. Linguist.* **36**(2), 247–277 (2010)
53. Wu, D.: Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Comput. Linguist.* **23**, 377–404 (1997)
54. Yamada, K., Knight, K.: A syntax-based statistical translation model. In: *Proceedings of the 39th Annual Meeting of the ACL*, Toulouse, France, pp. 523–530. ACL (2001)
55. Zollmann, A., Venugopal, A.: Syntax augmented machine translation via chart parsing. In: *Proceedings of the Workshop on Statistical Machine Translation*, New York, USA, pp. 138–141 (2006)