Data-Oriented Translation

Arjen Poutsma
Department of Computational Linguistics
University of Amsterdam
the Netherlands
poutsma@wins.uva.nl

Abstract

In this article, we present a statistical approach to machine translation that is based on Data-Oriented Parsing: Data-Oriented Translation (DOT). In DOT, we use linked subtree pairs for creating a derivation of a source sentence. Each linked subtree pair has a certain probability, and consists of two trees: one in the source language and one in the target language. When a derivation has been formed with these subtree pairs, we can create a translation from this derivation. Since there are typically many different derivations of the same sentence in the source language, there can be as many different translations for it. The probability of a translation can be calculated as the total probability of all the derivations that form this translation. We give the computational aspects for this model, show that we can convert each subtree pair into a productive rewrite rule, and that the most probable translation can be computed by means of Monte Carlo disambiguation. Finally, we discuss some pilot experiments with the Verbmobil corpus.

1 Introduction

The Data-Oriented Parsing model has been presented as a promising paradigm for natural language processing (Scha, 1990; Bod, 1995; Bod, 1998). It has been shown that DOP has the ability to locate syntactic and semantic dependencies, both of which are quite important for machine translation. We hope that, by basing our model on DOP, we can inherit these advantages, thus obtaining a new and interesting way to perform machine translation.

In section 2, we describe this novel model by identifying its parameters. In section 3, we describe its computational aspects; in section 4, we discuss some pilot experiments with this model; and finally, in section 5, we give some issues open for future research.

2 The Data-Oriented Translation Model

In this section, we will give the instantiation of a model that uses DOP for MT purposes, which we will call Data-Oriented Translation (DOT).[1] This model is largely based on DOP1 (Bod, 1998, chapt. 2).

In DOT, we use linked subtree pairs as combinational fragments.[2] Each linked subtree pair has a certain probability, and consists of a tree in the source language and a tree in the target language. By combining these fragments to form an an analysis of the source sentence, we automatically generate a translation, i.e. we form a derivation of both source sentence and target sentence. Since there are typically many different derivations which contain the same source sentence, there can be equally many different translations for it. The probability of a translation can be calculated as the total probability of all the derivations that form this translation.

The model presented here is capable of translating between two languages only. This limitation is by no means a property of the model itself, but is chosen for simplicity and readability reasons only.

The following parameters should be specified for a DOP-like approach to MT:

1. the *representations* of sentences that are assumed,
2. the *fragments* of these representations that can be used for generating new representations,
3. the *operator* that is used to combine the fragments to form a translation, and

[1] This is actually the second instantiation of such a framework. The original model (Poutsma, 1998; Poutsma, 2000) had a major flaw, which resulted in translations that were simply incorrect, as pointed out by Way (1999).

[2] Links between tree nodes were introduced for TAG trees, in (Schieber and Schabes, 1990), and put to use for Machine Translation by Abeillé et al. (1990).

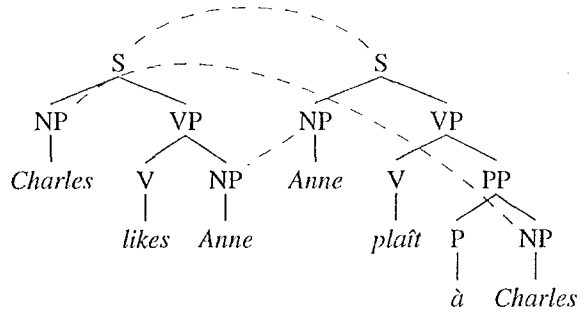

Figure 1: A linked tree pair $\langle T_s, T_t \rangle$.

4. the model that is used for determining the *probability* of a target sentence given a source sentence.

In the explanation that follows, we will use a subscript s to denote an element of the source language, and a subscript t to denote one of the target language.

2.1 Representations

In DOT, we basically use the same utterance-analysis as in DOP1 (i.e. syntactically labeled phrase structure trees). To allow for translation capabilities in this model, we will use pairs of trees that incorporate semantic information. The amount of semantic information need not be very detailed, since all we are interested in is semantic equivalence. Two trees T_1 and T_2 are said to be *semantic equivalents* (denoted as $T_1 \simeq T_2$) iff T_1 can be replaced with T_2 without loss of meaning.

We can now introduce the notion of *links*: a link symbolizes a semantic equivalence between two trees, or part of trees. It can occur at any level in the tree structure, except for the terminal level.[3]

The representation used in DOT is a 3-tuple $\langle T_s, T_t, \phi \rangle$, where T_s is a tree in the source language, T_t is a tree in the target language, and ϕ is a function that maps between semantic equivalent parts in both trees. In the rest of this article, we will refer to this 3-tuple as the pair $\langle T_s, T_t \rangle$.

Because of the semantic equivalence, a link must exist at the top level of the tree pair $\langle T_s, T_t \rangle$. Figure 1 shows an example of two linked trees, the links are depicted graphically as dashed lines.

[3]Links cannot occur at the terminal level, since we map between semantic equivalent parts on the level of syntactic categories.

2.2 Fragments

Likewise, we will use *linked subtrees* as our fragments. Given a pair of linked trees $\langle T_s, T_t \rangle$, a *linked subtree pair* of $\langle T_s, T_t \rangle$ consists of two connected and linked subgraphs $\langle t_s, t_t \rangle$ of $\langle T_s, T_t \rangle$ such that:

1. for every pair of linked nodes in $\langle t_s, t_t \rangle$, it holds that:
   (a) both nodes in $\langle t_s, t_t \rangle$ have either zero daughter nodes,
   or
   (b) both nodes have all the daughter nodes of the corresponding nodes in $\langle T_s, T_t \rangle$

   and
2. every non-linked node in either t_s (or t_t) has all the daughter nodes of the corresponding node in T_s (T_t),
   and
3. both t_s and t_t consist of more than one node.

This definition has a number of consequences. First of all, it is more restrictive than the DOP1 definition for subtrees, thus resulting in a smaller or equal amount of subtrees per tree. Secondly, it defines a *possible* pair of linked subtrees. Typically, there are many pairs of linked subtrees for each set of linked trees. Thirdly, the linked tree pair itself is also a valid linked subtree pair. Finally, according to this definition, all the linked subtree pairs are semantic equivalents, since the semantic daughter nodes of the original tree are removed or retained simultaneously (clause 1). The nodes for which a semantic equivalent does not exist are always retained (clause 2).

We can now define the *bag of linked subtree pairs*, which we will use as a grammar. Given a corpus of linked trees C, the *bag of linked subtree pairs* of C is the bag in which linked subtree pairs occur exactly as often as they can be identified in C.[4] Figure 2 show the bag of linked subtree pairs for the linked tree pair $\langle T_s, T_t \rangle$.

2.3 Composition operator

In DOT, we use the leftmost substitution operator for forming combinations of grammar rules. The composition of the linked tree pair $\langle t_s, t_t \rangle$ and

[4]The similarity between Example-based MT (Nagao, 1984) and DOT is clear: EBMT uses a database of examples to form a translation, whereas DOT uses a bag of structured trees.

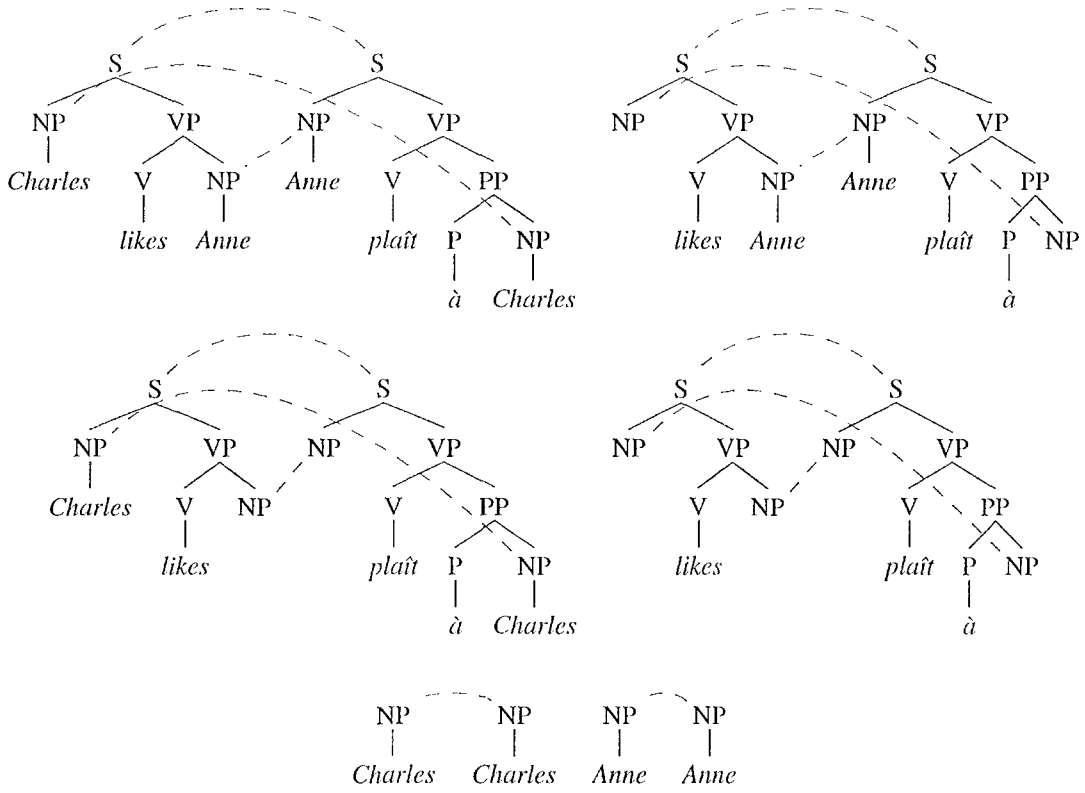

Figure 2: The bag of linked subtree pairs of $\langle T_s, T_t \rangle$

$\langle u_s, u_t \rangle$, written as $\langle t_s, t_t \rangle \circ \langle u_s, u_t \rangle$, is defined iff the label of the leftmost nonterminal linked frontier node and the label of its linked counterpart are identical to the labels of the root nodes of $\langle u_s, u_t \rangle$. If this composition is defined, it yields a copy of $\langle t_s, t_t \rangle$, in which a copy of u_s has been substituted on t_s's leftmost nonterminal linked frontier node, and a copy of u_t has been substituted on the node's linked counterpart. The composition operation is illustrated in figure 3.

Given a bag of linked subtree pairs B, a sequence of compositions $\langle t_{s_1}, t_{t_1} \rangle \circ \cdots \circ \langle t_{s_N}, t_{t_N} \rangle$, with $\langle t_{s_i}, t_{t_i} \rangle \in B$ yielding a tree pair $\langle T_s, T_t \rangle$ without nonterminal leaves is called a *derivation* D of $\langle T_s, T_t \rangle$.

2.4 Probability calculation

To compute the probability of the target composition, we make the same statistical assumptions as in DOP1 with regard to independence and representation of the subtrees (Bod, 1998, p. 16).

The probability of selecting a subtree pair $\langle t_s, t_t \rangle$ is calculated by dividing the frequency of the subtree pair in the bag by the number of subtrees that have the same root node labels in this bag. In other words, let $|\langle t_s, t_t \rangle|$ be the number of times the subtree pair $\langle t_s, t_t \rangle$ occurs in the bag of subtree pairs, and $r(t)$ be the root node categories of t, then the probability assigned to $\langle t_s, t_t \rangle$ is

$$P(\langle t_s, t_t \rangle) = \frac{|\langle t_s, t_t \rangle|}{\sum_{\langle u_s, u_t \rangle : r(u_s)=r(t_s) \wedge r(u_t)=r(t_t)} |\langle u_s, u_t \rangle|} \quad (1)$$

Given the assumptions that all subtree pairs are independent, the probability of a derivation $\langle t_{s_1}, t_{t_1} \rangle \circ \cdots \circ \langle t_{s_N}, t_{t_N} \rangle$ is equal to the product of the probabilities of the used subtree pairs.

$$P(\langle t_{s_1}, t_{t_1} \rangle \circ \cdots \circ \langle t_{s_N}, t_{t_N} \rangle) = \prod_i P(\langle t_{s_i}, t_{t_i} \rangle) \quad (2)$$

The translation generated by a derivation is equal to the sentence yielded by the target trees of the derivation. Typically, a translation can be generated by a large number of different derivations, each of which has its own probability. Therefore, the probability of a translation $w_s \Rightarrow w_t$ is the sum of the probabilities of its derivations:

$$P(w_s, w_t) = \sum P(D_{\langle w_s, w_t \rangle}) \quad (3)$$

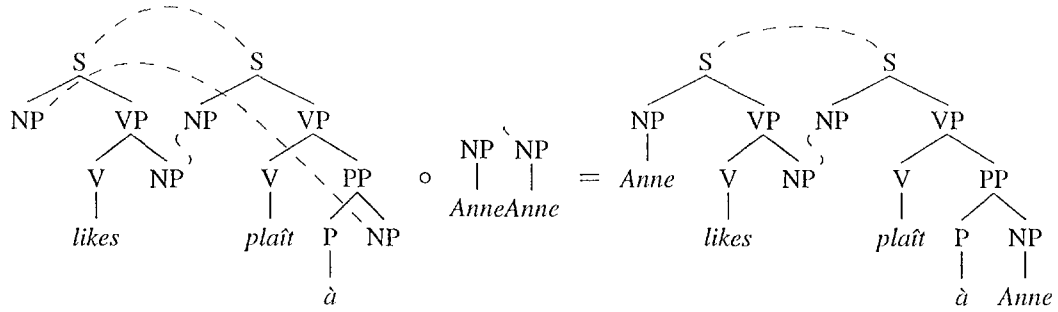

Figure 3: The composition operation

The justification of this last equation is quite trivial. As in any statistical MT system, we wish to choose the target sentence w_t so as to maximize $P(w_t|w_s)$ (Brown et al., 1990, p. 79). If we take the sum over all possible derivations that were formed from w_s and derive w_t, we can rewrite this as equation 4, as seen below. Since both w_s and w_t are contained in $D_{\langle w_s,w_t\rangle}$, we can remove them both and arrive at equation 5, which—as we maximize over w_t—is equivalent to equation 3 above.

$$\max_{w_t} P(w_t|w_s) =$$

$$= \max_{w_t} \sum_{D_{\langle w_s,w_t\rangle}} P(w_t, D_{\langle w_s,w_t\rangle}|w_s) \quad (4)$$

$$= \max_{w_t} \sum_{D_{\langle w_s,w_t\rangle}} P(D_{\langle w_s,w_t\rangle}) \quad (5)$$

3 Computational Aspects

When translating using the DOT model, we can distinguish between three computational stages:

1. *parsing*: the formation of a derivation forest,
2. *translation*: the transfer of the derivation forest from the source language to the target language,
3. *disambiguation*: the selection of the most probable translation from the derivation forest.

3.1 Parsing

In DOT, every subtree pair $\langle t_s, t_t\rangle$ can be seen as a productive rewrite rule: $\langle \text{root}(t_s), \text{root}(t_t)\rangle \rightarrow \langle \text{frontier}(t_s), \text{frontier}(t_t)\rangle$, where all linkage in the frontier nodes is retained. The linked non-terminals in the yield constitute the symbol pairs to which new rules (subtree pairs) are applied. For instance, the rightmost subtree pair in figure 3 can be rewritten as

$$\langle S, S\rangle \rightarrow \langle (\text{Anne}, \text{likes}, NP), (NP, \text{plaît}, \text{à}, \text{Anne})\rangle$$

This rule can then be combined with rules that have the root pair $\langle NP, NP\rangle$, and so on.

If we only consider the left-side part of this rule, we can use algorithms that exist for context-free grammars, so that we can parse a sentence of n words with a time complexity which is polynomial in n. These algorithms give as output a chart-like *derivation forest* (Sima'an et al., 1994), which contains the tree pairs of all the derivations that can be formed.

3.2 Translation

Since every tree pair in the derivation forest contains a tree for the target language, the translation of this forest is trivial.

3.3 Disambiguation

In order to select the most probable translation, it is not efficient to compare all translations, since there can be exponentially many of them. Furthermore, it has been shown that the Viterbi algorithm cannot be used to make the most probable selection from a DOP-like derivation forest (Sima'an, 1996).

Instead, we use a *random* selection method to generate derivations from the target derivation forest, otherwise known as Monte Carlo sampling (Bod, 1998, p. 46–49). In this method, the random choices of derivations are based on the probabilities of the underlying subderivations. If we generate a large number of samples, we can estimate the most probable translation as the translation which results most often. The most probable translation can be estimated as accurately as desired by making the number of random samples sufficiently large.

4 Pilot Experiments

In order to test the DOT-model, we did some pilot experiments with a small part of the Verbmobil corpus. This corpus consists of transliterated spoken appointment dialogues in German, English,

and Japanese. We only used the German and English datasets, which were aligned at sentence level, and syntactically annotated using different annotation schemes.[5]

Naturally, the tree pairs in the corpus did not contain any links, so—in order to make it useful for DOT—we had to analyze each tree pair, and place links where necessary. We also corrected tree pairs that were not aligned correctly. Figure 4 shows an example of a corrected and linked tree from our correction of the Verbmobil corpus.

We used a blind testing method, dividing the 266 trees of our corpus into an 85% training set of 226 tree pairs, and a 15% test set of 40 tree pairs. We carried out three experiments, in both directions, each using a different split of training and test set. The 226 training set tree pairs were converted into fragments (i.e. subtree pairs), and were enriched with their corpus probabilities. The 40 sentences from the test set served as input sentences: they were translated with the fragments from the training set using a bottom-up chart parser, and disambiguated by the Monte Carlo algorithm. The most probable translations were estimated from probability distributions of 1500 sampled derivations, which accounts for a standard deviation $\sigma \leq 0.013$. Finally, we compared the resulting translations with the original translation as given in the test set. We also fed the test sentences into another MT-system: AltaVista's Babelfish, which is based on Systran.[6]

4.1 Evaluation

In a manner similar to (Brown et al., 1990, p. 83), we assigned each of the resulting sentences a category according to the following criteria. If the produced sentence was exactly the same as the actual Verbmobil translation, we assigned it the *exact* category. If it was a legitimate translation of the source sentence but in different words, we assigned it the *alternate* category. If it made sense as a sentence, but could not be interpreted as a valid translation of the source sentence, we assigned it the *wrong* category. If the translation only yielded a part of the source sentence, we assigned it the *partial* category: either *partial exact* if it was a part of the actual Verbmobil translation, or *partial alternate* if it was part of an alternate translation. Finally, if no translation was given, we assigned it the *none* category. The results we obtained from Systran were also evaluated using this procedure. Figure 5 gives some classification examples.

Exact

	That would be very interesting.
Verbmobil:	Das wäre sehr interessant.
Translated as:	Das wäre sehr interessant.

Alternate

	I will book the trains.
Verbmobil:	Ich buche die Züge.
Translated as:	Ich werde die Züge reservieren.

Wrong

	Es ist ja keine Behörde.
Verbmobil:	It is not an administrative office you know.
Translated as:	There is not an administrative office you know.

Partial Exact

	And as said I think the location of the branch office is posh.
Verbmobil:	Und wie gesagt ich denke die Lage zur Filiale spricht Bände ist.
Translated as:	ich denke die Lage

Partial Alternate

	Ich habe Preise vom Parkhotel Hannover da.
Verbmobil:	I have got prices for Hannover Parkhotel here.
Translated as:	for Parkhotel Hannover

Figure 5: Translation and classification examples.

The method of evaluation is very strict: even if our model generated a translation that had a better quality than the given Verbmobil translation, we still assigned it the (partial) alternate category. This can be seen in the second example in figure 5.

4.2 Results

The results that we obtained can be seen in table 1 and 2. In both our experiments, the number of exact translations was somewhat higher than Systran's, but Systran excelled at the number of alternate translations. This can be explained by the fact that Systran has a much larger lexicon, thus allowing it to form much more alternate translations. While it is meaningless to compare results obtained from different corpora, it may be interesting to note that Brown et al. (1990) report a 5% exact match in experiments with the Hansard corpus, indicating that an exact match is very hard to achieve.

The number of ungrammatical translations in our

[5]The Penn Treebank scheme for English; the Tübingen scheme for German.

[6]This service is available on the Internet via `http://babelfish.altavista.com`.

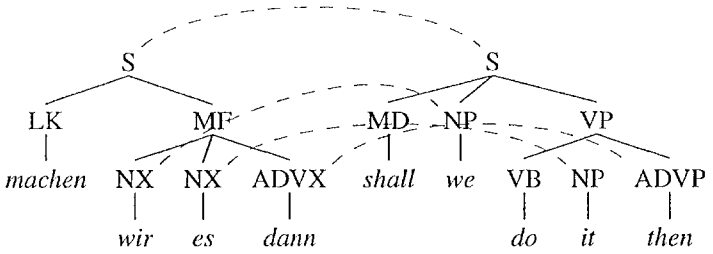

Figure 4: Example of a linked tree pair in Verbmobil

Corpus		Categorical accuracy					
Max. Depth	Size	Correct		Incorrect		Partial	
		Exact	Alternate	Ungr.	Wrong	Exact	Alternate
1	1263	16.22%	2.70%	18.92%	18.92%	18.92%	24.32%
2	2733	16.22%	2.70%	32.43%	5.41%	27.03%	16.22%
3	8228	18.92%	5.41%	32.43%	5.41%	24.32%	13.51%
4	14192	18.92%	5.41%	32.43%	5.41%	24.32%	13.51%
5	22147	18.92%	5.41%	32.43%	5.41%	24.32%	13.51%
6	27039	18.92%	5.41%	32.43%	5.41%	27.03%	10.81%
∞	33479	18.92%	5.41%	32.43%	5.41%	24.32%	13.51%
Systran		8.11%	37.84%	18.92%	35.14%	0%	0%

Table 1: Results of English to German translation experiments

English to German experiment were much higher than Systran's (32% versus Systran's 19%); vice-versa it was much lower (13% versus Systran's 21%). Since the German grammar is more complex than the English grammar, this result could be expected. It is simpler to map a complex grammar to a simpler than vice-versa.

The partial translations, which are quite useful for forming the basis of a post-edited, manual translation, varied around 38% in our English to German experiments, and around 55% when translating from German to English. Systran is incapable of forming partial translations.

As can be seen from the tables, we experimented with the maximum depth of the tree pairs used. We expected that the performance of the model would increase when we used deeper subtree pairs, since deeper structures allow for more complex structures, and therefore better translations. Our experiments showed, however, that there was very little increase of performance as we increased the maximum tree depth. A possible explanation is that the trees in our corpus contained a lot of lexical context (i.e. terminals) at very small tree depths. Instead of varying the maximum tree *depth*, we should experiment with varying the maximum tree *width*. We plan to perform such experiments in the future.

5 Future work

Though the findings presented in this article cover the most important issues regarding DOT, there are still some topics open for future research.

As we stated in the previous section, we wish to see whether DOT's performance increases as we vary the maximum width of a tree.

In the experiments it became clear that DOT lacks a large lexicon, thus resulting in less alternate translations than Systran. By using an external lexicon, we can form a part-of-speech sequences from the source sentence, and use this sequence as input for DOT. The resulting target part-of-speech sequence can then be reformed into a target sentence.

The experiments discussed in this article are pilot experiments, and do not account for much. In order to find more about DOT and its (dis)abilities, more experiments on larger corpora are required.

6 Conclusion

In this article, we have presented a new approach to machine translation: the Data-Oriented Translation model. This method uses linked subtree pairs for creating a derivation of a sentence. Each subtree-pair consists of two trees: one in the source language and one in the target language. Using these subtree pairs, we can form a derivation of a given source sentence, which can then be used to form a target sentence. The probability of a translation can

Corpus		Categorical accuracy					
Max. Depth	Size	Correct		Incorrect		Partial	
		Exact	Alternate	Ungr.	Wrong	Exact	Alternate
1	1263	15.38%	2.56%	12.82%	12.82%	41.03%	15.38%
2	2733	12.82%	7.69%	12.82%	12.82%	35.90%	17.95%
3	8228	12.82%	10.26%	12.82%	7.69%	38.46%	17.95%
4	14192	15.38%	7.69%	12.82%	10.26%	35.90%	17.95%
5	22147	15.38%	5.13%	12.82%	12.82%	35.90%	17.95%
6	27039	15.38%	5.13%	12.82%	10.26%	38.46%	17.95%
∞	33479	15.38%	7.69%	12.82%	7.69%	38.46%	17.95%
Systran		12.82%	25.64%	20.51%	41.03%	0%	0%

Table 2: Results of German to English translation experiments

then be calculated as the total probability of all the derivations that form this translation.

The computational aspects of DOT have been discussed, where we introduced a way to reform each subtree pair into a productive rewrite rule so that well-known parsing algorithms can be used. We determine the best translation by Monte Carlo sampling.

We have discussed the results of some pilot experiments with a part of the Verbmobil corpus, and showed a method of evaluating them. The evaluation showed that DOT produces less correct translation than Systran, but also less incorrect translations. We expected to see an increase in performance as we increased the depth of subtree pairs used, but this was not the case.

Finally, we supplied some topics which are open for future research.

References

A. Abeillé, Y. Schabes, and A.K. Joshi. 1990. Using lexicalized tags for machine translation. In *Proceedings of the 13th international conference on computational linguistics*, volume 3, pages 1–6, Helsinki.

R. Bod. 1995. *Enriching linguistics with statistics: Performance models of natural language*. Number 1995-14 in ILLC Dissertation Series. Institute for Logic, Language and Computation, Amsterdam.

R. Bod. 1998. *Beyond grammar: an experience-based theory of language*. Number 88 in CSLI lecture notes. CSLI Publications, Stanford, California.

J. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, J. Lafferty, R. Mercer, and P. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–86.

M. Nagao. 1984. A framework of a mechanical translation between Japanese and English by analogy principle. In A. Elithorn and R. Banerji, editors, *Artificial and Human Intelligence*, chapter 11, pages 173–180. North-Holland, Amsterdam.

A. Poutsma. 1998. Data-Oriented Translation. In *Ninth Conference of Computational Linguistics in the Netherlands*, Leuven, Belgium. Conference presentation.

A. Poutsma. 2000. Data-Oriented Translation: Using the DOP framework for MT. Master's thesis, Faculty of Mathematics, Computer Science, Physics and Astronomy, University of Amsterdam, the Netherlands.

R. Scha. 1990. Taaltheorie en taaltechnologie; competence en performance. In Q.A.M. de Kort and G.L.J. Leerdam, editors, *Computertoepassingen in de Neerlandistiek*. Landelijke Vereniging van Neerlandici, Almere, the Netherlands.

S.M. Schieber and Y. Schabes. 1990. Synchronous tree-adjoining grammars. In *Proceedings of the 13th international conference on computational linguistics*, volume 3, pages 253–258, Helsinki.

K. Sima'an, R. Bod, S. Krauwer, and R. Scha. 1994. Efficient disambiguation by means of stochastic tree substitution grammars. In *Proceedings International Conference on New Methods in Language Processing*, Manchester, UK. UMIST.

K. Sima'an. 1996. Computational Complexity of Probabilistic Disambiguation by means of Tree Grammars. In *Proceedings COLING-96*, Copenhagen, Denmark.

A. Way. 1999. A hybrid architecture for robust MT using LFG-DOP. *Journal of Experimental and Theoretical Artificial Intelligence*, 11(3).