Inducing Translation Grammars from Bracketed Alignments

Michael Carl
Institut für Angewandte Informationsforschung,
Martin-Luther-Straße 14
66111 Saarbrücken, Germany,
carl@iai.uni-sb.de

Abstract

This paper presents an algorithm for generating and filtering an invertible and structural analogous translation grammar from bilingual aligned and linguistically bracketed texts. The algorithm is discussed in general terms and applied to German-English alignments. It is shown that the induction of structural analogous translation grammars can lead to disambiguation of meaning and correction of bracketing errors.

1 Introduction

Recent developments of Example-Based Machine Translation (EBMT) show a trend of inducing translation grammars from aligned texts. The aligned text, which serves as a basis for the induction of translation grammars, consists of a set of alignments, i.e. a set of source language/target language expressions[1] which are translations of each other. A translation grammar, in turn, consist of lexical transfer rules and generalizations.

The advantage of translation grammars is 1) that they can be induced off-line thus reducing computation time in the working-phase of the system and 2) that consistency of the alignments can be checked and handled more easily.

According to (Somers, 1999), EBMT systems differ in the number and the quality of resources used and in the way this knowledge is represented, stored and used for translation. However, EBMT systems differ also in the way generalizations are computed. In the framework of EBMT, a number of methods have been proposed for inducing translation grammars. In so-called "pure" EBMT systems, the only available knowledge resource is the aligned text itself (cf. (Block, 2000; Brown, 2001)), while in richer systems linguistic knowledge resources are used to a varying degree (cf. (Güvenir and Cicekli, 1998) and this present approach). In almost all cases where generalizations are induced from aligned texts, a set of two or more alignments are compared and suitable sub-sequences are replaced by variables.

Grammar induction from sets of monolingual examples have been studied since the 1960s. As early as 1967, Gold showed that even regular grammars cannot be exactly identified from positive examples alone. This insight is based on the fact that there will always be at least two possible candidate hypotheses, i) the language contains all possible sentences over a set of symbols $\sum^*$ and ii) the language corresponds to the set of provided examples alone. While i) is most compact and general, ii) is the most complex and least general hypothesis. A similar situation can be constructed for the induction of translation grammars from alignments: i) each symbol in the target language is a possible translation of every symbol in the source language and ii) the translation grammar corresponds to the set of alignments alone. Neither of these hypotheses seem attractive and so it seems desirable to find a compromise between the size of the hypothesis description and their generality. The compromise for the induction of translation grammars has been to look for collocations in the left-hand side (LHS) and

[1] These expressions are typically sentences but can be single words, clauses or paragraphs.

Generalization of Differences in Alignments (Güvenir and Cicekli, 1998)

Alignments:

1.	I took a	ticket	from Mary	$\leftrightarrow$	Mary'den bir	bilet	aldim
2.	I took a	pen	from Mary	$\leftrightarrow$	Mary'den bir	kalem	aldim

Generalization of Differences:

3.	I took a	$\mathcal{X}_1$	from Mary	$\leftrightarrow$	Mary'den bir	$\mathcal{Y}_1$	aldim

---

Generalization of Differences and Generalization of Similarities in Alignments (McTait, 2001)

Alignments:

4.	The commission	gave	the plan	up	$\leftrightarrow$	La commission	abandonna	le plan
5.	Our government	gave	all laws	up	$\leftrightarrow$	Notre gouvernement	abandonna	toutes les lois

Generalization of Differences:

6.	(...)	gave	(...)	up	$\leftrightarrow$	(...)	abandonna	(...)

Generalization of Similarities:

7.	The commission	(...)	the plan	(...)	$\leftrightarrow$	La commission	(...)	le plan
8.	Our government	(...)	all laws	(...)	$\leftrightarrow$	Notre gouvernement	(...)	toutes les lois

---

Generalization of Chunk Pairs in Alignments (Block, 2000):

Alignment:

9.	das	ist	was	Sie	wollen am	Mittwoch	morgen	zurückzukommen
	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$		$\updownarrow$	$\updownarrow$	
	which	is	what	you	were wanting to come back	Wednesday	morning	

Chunk Pairs			Pattern Pairs						
das	$\leftrightarrow$	which							
ist	$\leftrightarrow$	is							
das ist	$\leftrightarrow$	which is	V ist	$\leftrightarrow$	V is	das V	$\leftrightarrow$	which V	
ist was	$\leftrightarrow$	is what	V was	$\leftrightarrow$	V what	ist V	$\leftrightarrow$	is V	
das ist was	$\leftrightarrow$	which is what	V ist was	$\leftrightarrow$	V is what	das V was	$\leftrightarrow$	which V what	...
ist was Sie	$\leftrightarrow$	is what you	V was Sie	$\leftrightarrow$	V what you	ist V Sie	$\leftrightarrow$	is V you	...
$\vdots$		$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$	

Figure 1: Methods for Inducing Generalizations from Alignments

right-hand side (RHS) of the alignments and to assume these collocations to be translated compositionally.

In this paper, I first examine previous work on how generalizations and translation grammars are computed from alignments. I then present an original approach which assumes sub-sequences in the LHS and RHS of the alignment to be bracketed. These brackets are potential candidates for compositional translation. From the bracketed sequences, the algorithm extracts hypotheses of transfer rules, generates generalizations and filters a translation grammar.

The algorithm is applied to German-English alignments. It is shown how the meaning of ambiguous sentences can be disambiguated by assuming structural analogy of both language sides and how bracketing errors can be corrected.

2 Previous Work

A translation grammar consists of lexical transfer rules and generalizations. Generalizations are generated while inducing a translation grammar and differ from lexical transfer rules as they contain variables.

With respect to the way generalizations are computed one can distinguish between i) methods which generalize chunk translations in alignments (e.g. (Block, 2000; Boström, 1999) and this present approach), ii) methods which generalize differences in sets of alignments (Güvenir and Cicekli, 1998) and iii) methods which generalize similarities in sets of alignments (McTait, 2001).

In (Güvenir and Cicekli, 1998) the system searches for pairs of alignments where the LHS and the RHS show similar sequences. The differences are then substituted by variables while the identical subsequences remain as non-reduced tokens in the generalizations. An example of this is given in figure 1, top. The identical parts in alignments 1 and 2 remain unreduced in generalization 3, while the different parts (*ticket* $\leftrightarrow$ *bilet* and *pen* $\leftrightarrow$ *kalem*) are reduced to a variable. (McTait, 2001) generalizes alignments in a similar way as shown in generalization 6 in figure 1, middle. In addition to this, (McTait, 2001) also generalizes similarities in alignments which yields generalization 5 and 6 as a complement to generalization 4. In contrast to the approach proposed in this paper, (McTait, 2001) allows $m : n$ mappings as, for instance, in generalization 6.

A different approach is taken by (Block, 2000). In his approach, word translations in alignments are assigned probabilities based on a statistical word alignment tool. Statistical word alignments are indicated by vertical flashes ($\updownarrow$) in the alignment 9, figure 1, bottom. Based on this statistical word translation information, so-called chunk pairs are extracted. Subsequently, generalizations (i.e. pattern pairs) are generated for each chunk pair. This is achieved by replacing a chunk pair in another chunk pair by a variable if the former is a substrings in the latter chunk pair. An example is given in figure 1, bottom. The chunk pairs *das ist* $\leftrightarrow$ *which is* and *das* $\leftrightarrow$ *which* have been extracted from the alignment 7. The pattern pair *V ist* $\leftrightarrow$ *V is* was obtained by replacing the shorter chunk pair by the variable V. As a last step, trivial pattern pairs are filtered from the set of pattern pairs.

3 Overview

The algorithm I present in this paper is similar to the approach in (Block, 2000). First, chunk pairs (i.e. lexical transfer rules) are extracted from each alignment. Then generalizations are computed from the extracted transfer rules and from the alignments. Last an invertible translation grammar is filtered. In contrast to (Block, 2000) the present algorithm does not use a statistical word alignment tool. Instead, constituents in both language sides of the alignments are bracketed. From these brackets, hypotheses of lexical transfer rules are extracted by combining each LHS-bracket with each RHS-bracket. We currently use an extended version of the shallow parser KURD (Carl and Schmidt-Wigger, 1998) for bracketing, although knowledge-poor methods could similarly be used as, for instance, described in (Zaanen, 2000).

Generalization works similarly to the model described in (Block, 2000), but more than one pattern pair can be replaced in one generalization. Also, probabilities and weights are assigned to each of the rules. From the set of extracted and generated rules, a compositional, invertible and structural analogous translation grammar is filtered. A translation grammar is structural analogous, if it generates structural analogous derivations for LHS and RHS. Struc-

c_1:(a)	$\leftrightarrow$	(e')	c_5:(e)	$\leftrightarrow$	(e')	c_9:	(de)	$\leftrightarrow$	(e')	c_{13}:(cde)	$\leftrightarrow$	(e')
c_2:(a)	$\leftrightarrow$	(a')	c_6:(e)	$\leftrightarrow$	(a')	c_{10}:	(de)	$\leftrightarrow$	(a')	c_{14}:(cde)	$\leftrightarrow$	(a')
c_3:(a)	$\leftrightarrow$	$(a'b')$	c_7:(e)	$\leftrightarrow$	$(a'b')$	c_{11}:	(de)	$\leftrightarrow$	$(a'b')$	c_{15}:(cde)	$\leftrightarrow$	$(a'b')$
c_4:(a)	$\leftrightarrow$	$(a'b'c')$	c_8:(e)	$\leftrightarrow$	$(a'b'c')$	c_{12}:	(de)	$\leftrightarrow$	$(a'b'c')$	c_{16}:(cde)	$\leftrightarrow$	$(a'b'c')$

Figure 2: Set of lexical transfer rules C_1 extracted from a_1

tural analogy is achieved if each generalization has the same number of variables in its LHS and RHS. This implies that any LHS derivation tree has the same depth and branching as the RHS derivation tree, while the order of the branches can be permuted in both language sides.

In addition, the filtered translation grammar does not contain ambiguous transfer rules, i.e. each LHS and RHS in the filtered translation grammar occurs exactly once. This invertibility condition is achieved by encoding a minimal context which makes it unique in the filtered translation grammar.

In the remainder of this paper, I first describe how transfer rules are extracted from bracketed alignments and how probabilities and weights are assigned. Then I describe how a set of invertible transfer rules is filtered. Last I give examples for inducing German-English translation grammars and discuss potentials of the algorithm.

4 Complexity of Inducing Translation Templates

The program assumes a bilingual text P which consists of n alignments $a_1 \ldots a_n$. Each alignment a_i consists of a left-hand side (LHS) and a right-hand side (RHS). By means of a (shallow) parser, both language sides are independently bracketed (parsed) which results in a representation similar to the following[2]:

$$a_1 : (a)\ b\ (c(d(e))) \leftrightarrow (((a')b')c')\ d'\ (e')$$

Without further knowledge, one cannot know which of the LHS-brackets translate into its corresponding bracket in the RHS or whether a LHS-bracket has a RHS-translation at all. But it is assumed that if a LHS-bracket has a translation in the RHS of the same alignment then it would translate into exactly one bracket. As we have no knowledge which brackets to align, we assume that each LHS-bracket translates with the same probability into any RHS-bracket. For an alignment a_i we can therefore extract $p \times q$ lexical transfer rules $C_i : \{c_1 \ldots c_{p*q}\}$, where p is the number of LHS-brackets and q is the number of RHS-brackets. The set C_1 extracted from a_1 is shown in figure 2.

In a second step, translation templates (or generalizations) are induced from the alignments and from the extracted lexical transfer rules $c_1 \ldots c_{p*q}$. While a lexical transfer rule consists only of terminal symbols, a generalization contains variables (so-called reductions) in places where a shorter transfer rule matches a sub-sequence in the LHS and in the RHS. A generalization has thus at least one reduction in each language side and it has an equal number of reductions in the LHS and the RHS. As we assume a 1-to-1 mapping of the LHS and RHS-brackets each reduction in LHS is linked to exactly one reduction in RHS and vice versa.

From the transfer rule c_{16}, for instance, can be generated 4 different generalizations while from transfer rule c_{11} only one generalization can be generated. These generalizations are shown in figure 3. From the alignment a_1 can be generated 25 generalizations by substituting one or more transfer rules $c_1 \ldots c_{16}$.

More generally, from a transfer rule c_j which has p brackets in its LHS and q brackets in its RHS, an exponential number of generalizations $\#G_{c_j}$ can be generated:

$$\#G_{c_j} = \sum_{i=0}^{p} \binom{p}{i} * \binom{q}{i} = \binom{p+q}{p}$$

For instance, if we assume both, p and q to be 10 and none of the 10 brackets in either language side are included in another bracket (i.e. all brackets are top-level brackets), more than 180,000 different generalizations can be generated. This is a number far too big to be computed, as 10 brackets (i.e. constituents in the parsed sentence) is not many. Therefore,

[2]The letters "$abcde$" represent lemmas in LHS; the letters "$a'b'c'd'e'$" lemmas in the RHS. The brackets are also annotated with a phrasal tag and morpho-syntactic information. For the sake of simplicity, I will consider this information only in the last two sections.

$$
\begin{array}{lllll}
G_{c_i} & \text{Induced Generalization} & & p(g_k) & w(g_k) \\
G_{c_{11}}: & \{\ g_1: \quad (dA) & \leftrightarrow \quad (A'b') & 0.2 & 0.4 \\
\\
G_{c_{16}}: & \left\{ \begin{array}{lll}
g_2: & (cdA) & \leftrightarrow \quad (A'b'c') \\
g_3: & (cdA) & \leftrightarrow \quad (A'c') \\
g_4: & (cA) & \leftrightarrow \quad (A'b'c') \\
g_5: & (cA) & \leftrightarrow \quad (A'c')
\end{array} \right. & & \begin{array}{ll} 0.2 & 0.4 \\ 0.2 & 0.4 \\ 0.2 & 0.4 \\ 0.2 & 0.6 \end{array}
\end{array}
$$

Figure 3: Set of generalizations $G_{c_{11}}$ and $G_{c_{16}}$ induced from transfer rules c_{11} and c_{16}

a number of heuristics is proposed for generating from the set of possible generalizations only those achieving the highest weight.

5 Probabilistic Translation Grammars

Before introducing heuristics, I would like to describe how probabilities and weights are assigned to lexical transfer rules and generalizations[3]. While inducing the translation grammar, each alignment a_i, each lexical transfer rule c_j and each generalization g_k, is linked to two sets.

$$
\begin{array}{rcl}
a_i & \Rightarrow & \{G_{a_i}, C_{a_i}\} \\
g_k & \Rightarrow & \{R_{g_k}, C_{g_k}\} \\
c_j & \Rightarrow & \{G_{c_j}, A_{c_j}\}
\end{array}
$$

Alignments a_i are associated with a set of lexical transfer rules C_{a_i} and a set of generalizations G_{a_i}. The lexical transfer rules in C_{a_i} have been extracted from alignment a_i while G_{a_i} contains generalizations of a_i. Each lexical transfer rule c_j is associated with a set A_{c_j} of alignments from which c_j has been extracted and a set of generalizations G_{c_j} which have been generated from c_j. Finally, each generalization g_k is associated with a set of lexical transfer rules C_{g_k} which have been replaced in the generalization (i.e. the daughters of the generalization g_k) and a set of references R_{g_k}. Since generalizations are generated from alignments and from the extracted lexical transfer rules, the set of references R_{g_k} may consist of alignments a_i and/or transfer rules c_j.

The probability of an alignment a_i is its frequency in the aligned text P divided by the number n of alignments in P.

[3]The notation used here is slightly different from a previous presentation in (Carl, 2001).

$$
p(a_i) = \frac{f(a_i)}{n} \tag{1}
$$

The probability of a lexical transfer rule c_j is a function of the number of times c_j has been extracted from alignments a_i, $i = 1 \ldots n$, the cardinality $\#C_{a_i}$ of the set C_{a_i} and the size of P:

$$
p(c_j) = \sum_{a_i \in A_{c_j}} \frac{1}{\sqrt{\#C_{a_i}} + n} \tag{2}
$$

The probability of a generalization g_k equals the maximum probability of the reference $r \in R_{g_k}$ from which g_k has been generated.

$$
p(g_k) = max\{p(r \in R_{g_k})\} \tag{3}
$$

Based on these probabilities, a weight is computed for each a_i, c_j, g_k.

The weight $w(r_j)$ of a lexical transfer rule or an alignment equals the maximum weight of the generalization that has been generated from it.

$$
w(r_j) = max\{w(g \in G_{r_j})\} \tag{4}
$$

In case no generalization can be generated from r_j: $w(r_j) = p(r_j)$. The weight of a generalization g_k equals the maximum of the probability of its reference $r_i \in R_{g_k}$ plus the sum of the weights of the lexical transfer rules which have been replaced in r_i to generate g_k. (i.e. $p(r_i)$ plus the sum of the daughters of g_k for reference r_i).

$$
w(g_k) = max\{p(r_i \in R_{g_k}) + \sum_{c_i \in C_{g_k}} w(c_i)\} \tag{5}
$$

From this it follows that generalizations have higher weights if they contain i) more reductions or ii) if the reductions are to the highest extend compositional. As an example for achieving higher weights for more compositional generalizations, consider the following example.

As there are 16 lexical transfer rules extracted from alignment a_1 and assuming that $n = 1$, each of the rules in figure 2 has probability 0.2. A sub-sequence in $c_{11} : (de) \leftrightarrow (a'b')$ can be substituted by rule $c_6 : (e) \leftrightarrow (a')$. This substitution yields generalization g_1, as shown in figure 3. This generalization[4] is assigned the weight 0.4 (cf. equation 5). The same weight is assigned to the transfer rule c_{11} (cf. equation 4). Subsequently, when generalizing the longer rule $c_{16} : (cde) \leftrightarrow (a'b'c')$, a set of four generalizations G_{16} is generated from which g_5 is assigned the highest weight due to the compositional nature of the replaced daughter $c_{11} : (de) \leftrightarrow (a'b')$.

6 Generating Transfer Rules

Aligned texts $P : \{a_1 \ldots a_n\}$ consist of $n \geq 1$ alignments. Each alignments $a_i,\ i = 1 \ldots n$ is generalized in a sequential manner. For each a_i, first the $p \times q$ lexical transfer rules are extracted and sorted by the length of the shorter string LHS or RHS. Transfer rules $c_j \in C_{a_i}$ are then generalized starting with the shortest rule. The crucial points in the procedure **Generate-Grammar**() are lines 4 and 9. Extracting the set of lexical transfer requires a quadratic effort in the number of brackets, while generating G_{c_j} from c_j is exponential. To tackle this latter complexity, a version of the A* algorithm generates only a limited number of the highest weighted generalization:

```
1  GenerateGrammar(P)
2  begin
3    for all a_i ∈ P:
4      extract lexical transfer rules C_{a_i} from a_i;
5      for all c_j ∈ C_{a_i} : p(c_j) += 1/(sqrt(#C_{a_i}) + n);
6      add a_i as c_0 to C_{a_i};
7      sort C_{a_i} by length of shorter LHS or RHS;
8      for each c_j ∈ C_{a_i} starting with shortest c
9        generate G_{c_j} from c_j
10       w(c_j) = max{w(g ∈ G_{c_j})}
11     end
12   end
13 end;
```

[4]The capital letters A, A', B and B' represent reductions of the replaced substrings where A maps into A' and B maps into B'.

In this way generalization of lexical transfer rules is reduced to $O(k * d)$ where k is the number of generalizations to be generated from c_j and d is the number of daughters in a generalization. In addition to this, a couple of parameters can be set to reduce the number of extracted transfer rules and generalizations:

- Transfer rules are only extracted where the difference in the number of words in LHS and RHS does not exceed a certain threshold. This constraint reflects the fact that generally the same number of (content) words appear on both sides of a translation.
- By the same token, transfer rules and generalizations are weighted by the difference of number of words in LHS and RHS.
- By means of a bilingual lexicon, transfer rules and generalizations may be assigned (high) priori weights.
- Only a limited number of highest weighted translation rules and generalizations is generated.
- A generalization can have a maximum number of reductions.
- Transfer rules are only asserted if their weights are above a threshold of an already existing, ambiguous transfer rule in the database.

Note that these parameters are suitable to reduce the number of generated generalizations and extracted lexical transfer rules.

7 Filtering Invertible Translation Grammars

From the set of alignments $P : \{a_1, \ldots, a_n\}$ and their associated sets of generalizations $\{G_{a_1}, \ldots, G_{a_n}\}$ an invertible translation grammar is filtered in a top down fashion. The aim here is to find a compositional set of transfer rules capable of reproducing the aligned text P in a most complete manner. To achieve this goal, translation ambiguities in the resulting grammar are avoided by including the smallest possible context which disambiguates each transfer rule. A transfer rule $\text{LHS}_2 \leftrightarrow \text{RHS}_2$

Default Translation Grammar

			$p(\cdot)$	$w(\cdot)$
e	$\leftrightarrow$	a'	0.2	0.2
a	$\leftrightarrow$	e'	0.2	0.2
dA	$\leftrightarrow$	$A'b'$	0.2	0.4
cA	$\leftrightarrow$	$A'c'$	0.2	0.6
AbB	$\leftrightarrow$	$A'd'B'$	1.0	1.8

Translation Grammar with Prior Lexical Knowledge

prior transfer rule $p(a \leftrightarrow a') = 0.5$

			$p(\cdot)$	$w(\cdot)$
a	$\leftrightarrow$	a'	0.50	0.50
e	$\leftrightarrow$	e'	0.24	0.24
$AbcdB$	$\leftrightarrow$	$A'b'c'd'B'$	1.00	1.74

Figure 4: Translation Grammar induced from alignment $(a)\ b\ (c(d(e))) \leftrightarrow (((a')b')c')\ d'\ (e')$

is ambiguous iff the translation grammar contains a different transfer rule $\mathrm{LHS}_1 \leftrightarrow \mathrm{RHS}_1$ where either LHS_1 equals LHS_2 or RHS_1 equals RHS_2. A translation grammar is invertible iff it contains no ambiguous transfer rules. In an invertible translation grammar, therefore, each LHS-string and each RHS-string occurs exactly once. Note that an invertible translation grammar is not deterministic. More than one translation can be generated from a source language string by differently bracketing the source string. However, each bracket has exactly one translation.

The alignments $a_i \in P$ are sorted by their weights and for each alignment starting with the highest weighted one, the function **FilterGrammar**(a_i) is called. The procedure **FilterGrammar()** prints the highest weighted generalizations of a_i and recursively prints their daughters. Less frequent rules, i.e. lower weighted rules, are likely to come along with more context. Only one generalization is printed for each alignment.

```
1  FilterGrammar(r_i)
2  begin
3    if G_{r_i} not empty
4      print g_k : max{w(g_k ∈ G_{r_i})}
5      for all generalizations g_m : delete g_m \
         if((LHS_m = LHS_k) or (RHS_m = RHS_k)).
6      for all c_j ∈ C_{g_k}: FilterGrammar(c_j)
7    else
8      print r_i
9      for all rules c_m: delete c_m \
         if((LHS_m = LHS_i or (RHS_m = RHS_i)).
10   end
11 end;
```

The procedure is called recursively in line 6 to print the highest weighted daughters c_j of generalization g_k.

Without prior knowledge of translation equivalences, the algorithm filters the most compositional, structural analogous translation grammar. A maximal compositional translation grammar generated and filtered from alignment a_1 is shown in figure 4, left. However, if the program is fed with prior knowledge of translation equivalences which do not - or only partially - support the structural equivalence of the bracketed alignments, a different translation grammar might be generated. Thus, the translation grammar in figure 4, right, is generated when providing the transfer rule $a \leftrightarrow a'$ with a prior probability of $p(a \leftrightarrow a') = 0.5$. Instead of the 16 lexical transfer rules in figure 2, only 10 transfer rules are extracted from alignment a_1 as $\{c_1, c_3, c_4, c_6, c_{10}, c_{14}\}$ are not consistent with the prior knowledge. The program tries to take this knowledge into account by filtering the most compositional translation grammar which is consistent with the data.

8 Ambiguous Bracketing

It is not always possible to obtain unambiguous brackets for every alignment. Ambiguous brackets might be due to (mal-) performance of the bracketing tool or to the implicit ambiguity of the sentence.

In the worst case, every subsequence in an alignment can be bracketed. There are maximally $\sum_{i=1}^{n} i$ different brackets in a sentence of length n. For an alignment with n tokens in LHS and m tokens in RHS, the number of extracted transfer rules amounts, thus, to:

$$\#C \ = \ \frac{(n^2+n)*(m^2+m)}{4}$$

While there are many different possible translation grammars that could be generated and filtered from this set, the most compositional grammar is very simple.

An example is given below. Assume all proper subsequences in the LHS and RHS of alignments a_2 and a_3 in figure 5 are bracketed. As both sides of the alignments have

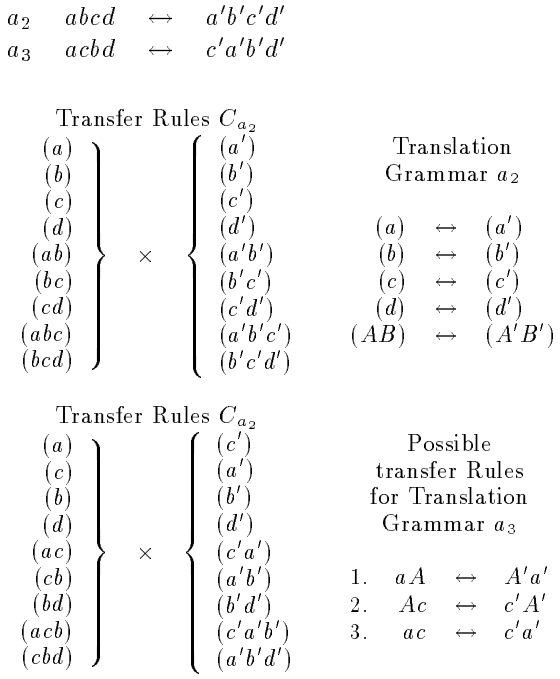

Figure 5: Maximal Ambiguous Bracketing and Maximal Compositional Translation Grammar

four token and assuming that the entire alignments are not bracketed, this leads to $\#C_{a_2} = 9 \times 9 = 81$ transfer rules for alignment a_2 and $\#C_{a_3} = 9 \times 9 = 81$ transfer rules for alignment a_3. 16 of the transfer rules in C_{a_2} and C_{a_3} are identical such that in total 146 different transfer rules are generated[5]. In Figure 5 the LHS and RHS-brackets are plotted from which the 146 translation rules are generated. These rules and the alignments are then generalized and a translation grammar is filtered as described above. The most compositional translation grammar filtered for alignment a_2 consists of 5 translation rules as shown in figure 5, top right. For alignment a_3 only one additional translation rule is filtered. There are many possibilities to adding one transfer rule to translation grammar a_2 such that the grammar is still invertible and alignment a_3 can be generated. Three possible transfer rules are shown in figure 5 bottom, right.

From an invertible translation grammar, different translations may be obtained by applying a different subset and/or changing the order of transfer rules. Adding rule 2 to translation grammar a_2 increases the number of possible derivations which can be generated from $a'b'c'd'$ and, as a consequence of this, increases the number of possible translations:

$$(a')(b')(c')(d') \rightarrow abcd$$
$$(a')(b')(c'(d')) \rightarrow abdc$$

While bracketing a sentence in a different way may result in different translations for that sentence, in an invertible translation grammar each bracket and thus each transfer rule has one unique translation. To achieve higher reliability of the induced grammar, the function **FilterGrammar()** is, therefore, tuned to filter transfer rule 3 which reduces this kind of non-determinism.

9 Disambiguating Meaning

In the remaining part of this paper I investigate a few implications of the algorithm applied to German-English alignments. In this section I show that the meaning of an alignment can be disambiguated by filtering a structural analogous translation grammar. In the next section I show that the algorithm can also correct wrong brackets in alignments.

Consider the German-English alignment in figure 6. While there is only one non-overlapping sequence of brackets in the German LHS, the English translation has two interpretations. From the English sentence one cannot know whether *the block* is in *the box* or whether *the box* is on *the table*. Accordingly, two ambiguous brackets are generated: *(the block in the box)* and *(the box on the table)* which express these two interpretations. In the German translation, the relations of the phrases (i.e. the attachment of the PPs) is clarified by their cases. The dative *in der Kiste* expresses the location of the *Klotz*, while the accusative of *auf den Tisch* expresses the direction where the *Klotz* is placed. Assuming the same meaning of the German and the English sentences, *the block* is, thus, in *the box* and *John puts the block on the table*.

As discussed above, the algorithm filters transfer rules which show best structural analogy of the English and German brackets. When generating the translation grammar, the meaning of the English sentence is disambiguated ac-

[5]Note, however, that the program indexes each LHS and RHS only once, while the connections between LHS and RHS are realized through pointer. This amounts to 14 indexes in each language side of $C_{a_2 \cup a_3}$ and 2 * 146 connecting pointer.

$(Hans)_{\texttt{noun}}$ *stellt* $((den\ Klotz)_{\texttt{dp}}$ *in* $(der\ Kiste)_{\texttt{dp}})_{\texttt{dp}}$ *auf* $(den\ Tisch)_{\texttt{dp}}$

$\leftrightarrow$

$(John)_{\texttt{noun}}$ *puts* $((the\ block)_{\texttt{dp}}$ *in* $(the\ box)_{\texttt{dp}})_{\texttt{dp}}$ *on* $(the\ table)_{\texttt{dp}}$

$(John)_{\texttt{noun}}$ *puts* $(the\ block)_{\texttt{dp}}$ *in* $((the\ box)_{\texttt{dp}}$ *on* $(the\ table)_{\texttt{dp}})_{\texttt{dp}}$

Filtered Translation Grammar			$p(\cdot)$	$w(\cdot)$
$(Tisch)_{\texttt{noun}}$	$\leftrightarrow$	$(table)_{\texttt{noun}}$	0.50	0.50
$(Kiste)_{\texttt{noun}}$	$\leftrightarrow$	$(box)_{\texttt{noun}}$	0.10	0.10
$(Klotz)_{\texttt{noun}}$	$\leftrightarrow$	$(block)_{\texttt{noun}}$	0.10	0.10
$(Hans)_{\texttt{noun}}$	$\leftrightarrow$	$(John)_{\texttt{noun}}$	0.10	0.10
$(art\ \{\texttt{noun}\}^1)_{\texttt{dp}}$	$\leftrightarrow$	$(the\ \{\texttt{noun}\}^1)_{\texttt{dp}}$	0.10	0.60
$(\{\texttt{dp}\}^1\ in\ \{\texttt{dp}\}^2)_{\texttt{noun}}$	$\leftrightarrow$	$(\{\texttt{dp}\}^1\ in\ \{\texttt{dp}\}^2)_{\texttt{noun}}$	0.10	1.30
$(\{\texttt{noun}\}^1\ stellt\ \{\texttt{dp}\}^2\ auf\ \{\texttt{dp}\}^3)$	$\leftrightarrow$	$(\{\texttt{noun}\}^1\ puts\ \{\texttt{dp}\}^2\ on\ \{\texttt{dp}\}^3)$	1.00	3.00

Figure 6: Disambiguation of PP attachment

cording to the brackets of the German translation. The filtered translation grammar is shown in figure 6. Here we reproduce an experience of (Melamed, 2001):
"... the translation of a text into another language can be viewed as a detailed annotation of what that text means".
In this example the translation leads to a disambiguation of the English PP-attachment.

Note that the translation $(Tisch)_{\texttt{noun}} \leftrightarrow (table)_{\texttt{noun}}$ was provided with a prior probability of 0.5. The curely brackets in the translation grammar represent reductions previously marked by the capital letters A and B. The features[6] {noun} and {dp} are phrasal tags of the transfer rules and brackets which also appear in reductions of generalizations. The superscriped indices 1, 2 and 3 in the LHS and RHS-reductions denote the linking of the reductions. Note also, that the algorithm works on sequences of lemmas, rather than on the surface forms of the words. The lemma *art* in the German side of the transfer rule represents the surface forms of the articles *der, die, das, dem, den, des.*

Disambiguation of meanings through structural analogy does not work in cases where both language sides are equally ambiguous as, for instance, in the famous example:

Hans sieht den Mann mit dem Fernglas $\leftrightarrow$
John sees the man with the binoculars

[6]Nouns and adjective-noun combinations are marked noun, simple and complex determiner phrases are marked dp.

1. $(\{\texttt{noun}\}^1\ sieht\ \{\texttt{dp}\}^2) \leftrightarrow$
   $(\{\texttt{noun}\}^1\ sees\ \{\texttt{dp}\}^2)$
2. $(\{\texttt{noun}\}^1\ sieht\ \{\texttt{dp}\}^2\ \{\texttt{pp}\}^3) \leftrightarrow$
   $(\{\texttt{noun}\}^1\ sees\ \{\texttt{dp}\}^2\ \{\texttt{pp}\}^3)$

Given the alignment is appropriately bracketed, at least two top-level generalizations are generated.

In case there is no conflicting evidence in the aligned text, the program is likely to filter the more compositional generalization 1 which assumes an attachment of the PPs to the close-by object rather than an attachment to the subject as in generalization 2.

10 Correcting Bracketing Errors

Meaning of sentences and their corresponding brackets can be ambiguous as shown above. Bracketing can also be be misguiding or wrong. However, if most of the brackets in a set of alignments are correct, the algorithm is likely to extract a correct translation grammar. Wrong brackets in alignments are ignored when filtering a compositional and structural analogous translation grammar from the alignments.

In the alignments in figure 7, only nouns and adjective-noun combinations (noun) as well as simple and complex determiner phrases (dp) are bracketed. While most of the brackets are correct, some of them are linguistically not justified. These unjustified brackets are purposely added to see how far the grammar induction can tackle such cases.

In alignments 1 and 6 all brackets in the German and English side are correct. Alignment 1 has a bracketed noun and one simple determiner phrase in either language side, while in alignment 6, complex embedded phrases are

1. *(Peter)*$_{\text{noun}}$ *lässt (das (Haus)*$_{\text{noun}}$*)*$_{\text{dp}}$ *ungern unbewohnt.* $\leftrightarrow$
   (Peter)$_{\text{noun}}$ *doesn't like (the (house)*$_{\text{noun}}$*)*$_{\text{dp}}$ *to be left empty.*

2. *Man hat ((Peter)*$_{\text{noun}}$ *(das (Rad)*$_{\text{noun}}$*)*$_{\text{dp}}$*)*$_{\text{noun}}$ *wechseln sehen.* $\leftrightarrow$
   (Peter)$_{\text{noun}}$ *was seen changing (the (wheel)*$_{\text{noun}}$*)*$_{\text{dp}}$*.*

3. *((Peter)*$_{\text{noun}}$ *(der (Wolf)*$_{\text{noun}}$*)*$_{\text{dp}}$*)*$_{\text{noun}}$ *hält (ein (Referendum)*$_{\text{noun}}$*)*$_{\text{dp}}$ *für überflüssig.* $\leftrightarrow$
   ((Peter)$_{\text{noun}}$ *(the (wolf)*$_{\text{noun}}$*)*$_{\text{dp}}$*)*$_{\text{noun}}$ *thinks (a (referendum)*$_{\text{noun}}$*)*$_{\text{dp}}$ *unnecessary.*

4. *Man hat ((Peter)*$_{\text{noun}}$ *in (das (Haus)*$_{\text{noun}}$*)*$_{\text{dp}}$*)*$_{\text{noun}}$ *gehen sehen.* $\leftrightarrow$
   (Peter)$_{\text{noun}}$ *was seen to enter (the (house)*$_{\text{noun}}$*)*$_{\text{dp}}$*.*

5. *Es ist bekannt, dass ((Peter)*$_{\text{noun}}$ *(ein (Lügner)*$_{\text{noun}}$*)*$_{\text{dp}}$*)*$_{\text{noun}}$ *ist.* $\leftrightarrow$
   (Peter)$_{\text{noun}}$ *is known to be (a (liar)*$_{\text{noun}}$*)*$_{\text{dp}}$*.*

6. *Von (Peter)*$_{\text{noun}}$ *wird erwartet, dass er mit ((den (Gesetzen)*$_{\text{noun}}$*)*$_{\text{dp}}$ *(seines (eigenen (Landes)*$_{\text{noun}}$*)*$_{\text{noun}}$*)*$_{\text{dp}}$*)*$_{\text{dp}}$ *vertraut ist.* $\leftrightarrow$
   (Peter)$_{\text{noun}}$ *is supposed to know ((the (laws)*$_{\text{noun}}$*)*$_{\text{dp}}$ *of (his (own (country)*$_{\text{noun}}$*)*$_{\text{noun}}$*)*$_{\text{dp}}$*)*$_{\text{dp}}$*.*

$p(\cdot)$	$w(\cdot)$			
		Translation Grammar filtered from Alignment a_3		
0.14	1.48	({noun}^{1} hält {dp}^{2} für überflüssig.)	$\leftrightarrow$	({noun}^{1} thinks {dp}^{2} unnecessary.)
0.11	0.22	(ein {noun}^{1})$_{\text{dp}}$	$\leftrightarrow$	(a {noun}^{1})$_{\text{dp}}$
0.11	0.11	(Referendum)$_{\text{noun}}$	$\leftrightarrow$	(referendum)$_{\text{noun}}$
0.08	1.12	({noun}^{1} {dp}^{2})$_{\text{noun}}$	$\leftrightarrow$	({noun}^{1} {dp}^{2})$_{\text{noun}}$
0.22	0.44	(art {noun}^{1})$_{\text{dp}}$	$\leftrightarrow$	(the {noun}^{1})$_{\text{dp}}$
0.08	0.08	(Wolf)$_{\text{noun}}$	$\leftrightarrow$	(wolf)$_{\text{noun}}$
0.60	0.60	(Peter)$_{\text{noun}}$	$\leftrightarrow$	(Peter)$_{\text{noun}}$
		Translation Grammar filtered from Alignment a_6		
0.13	1.47	Von {noun}^{1} wird erwartet, dass er mit {dp}^{2} vertraut ist.)	$\leftrightarrow$	({noun}^{1} is supposed to know {dp}^{2}.)
0.07	0.75	({dp}^{1} {dp}^{2})$_{\text{dp}}$	$\leftrightarrow$	({dp}^{1} of {dp}^{2})$_{\text{dp}}$
0.08	0.08	(Gesetz)$_{\text{noun}}$	$\leftrightarrow$	(law)$_{\text{noun}}$
0.08	0.23	(sein {noun^1})$_{\text{dp}}$	$\leftrightarrow$	(his {noun^1})$_{\text{dp}}$
0.08	0.15	(eigen {noun}^{1})$_{\text{noun}}$	$\leftrightarrow$	(own {noun}^{1})$_{\text{noun}}$
0.08	0.08	(Land)$_{\text{noun}}$	$\leftrightarrow$	(country)$_{\text{noun}}$
		Translation Grammar filtered from Alignment a_4		
0.15	1.22	(Man hat {noun}^{1} in {dp}^{2} gehen sehen.)	$\leftrightarrow$	({noun}^{1} was seen to enter {dp}^{2}.)
0.22	0.22	(Haus)$_{\text{noun}}$	$\leftrightarrow$	(house)$_{\text{noun}}$
		Translation Grammar filtered from Alignment a_2		
0.15	1.21	(Man hat {noun}^{1} {dp}^{2} wechseln sehen.)	$\leftrightarrow$	({noun}^{1} was seen changing {dp}^{2}.)
0.11	0.11	(Rad)$_{\text{noun}}$	$\leftrightarrow$	(wheel)$_{\text{noun}}$
		Translation Grammar filtered from Alignment a_1		
0.13	1.20	({noun}^{1} lässt {dp}^{2} ungern unbewohnt.)	$\leftrightarrow$	({noun}^{1} doesn't like {dp}^{2} to be left empty.)
		Translation Grammar filtered from Alignment a_5		
0.14	0.99	(Es ist bekannt, dass {noun}^{1} {dp}^{2} ist.)	$\leftrightarrow$	({noun}^{1} is known to be {dp}^{2}.)
0.11	0.11	(Lügner)$_{\text{noun}}$	$\leftrightarrow$	(liar)$_{\text{noun}}$

Figure 7: Wrong Brackets in Alignments and Correct Translation Grammar

bracketed. Note that the structure of the brackets in the LHS and RHS of these alignments are identical.

In alignment 2, the bracket *(Peter das Rad)* is wrong as *Peter* is the object of the verb *sehen* and in the same time the subject of *wechseln* while *das Rad* is the accusative object of *wechseln*. The problem here is that *das Rad* is ambiguous wrt. nominative or accusative. Similar brackets in alignment 3, *(Peter der Wolf)* are correct as the nominative *der Wolf* modifies the subject *Peter*.

The distinction between these two cases is difficult to determine for a partial parser as it requires knowledge about valency of verbs and the type of the sentence (i.e. verb middle or verb final). However, if we look at the English brackets of these alignments, one sees that *Peter* and *the wheel* are discontinuous in alignment 2 while the bracket *(Peter the wolf)* in alignment 3 show the same structure as German *(Peter der Wolf)*. Since the algorithm tries to filter most compositional, structural analogous translation grammars, the wrong German bracket in alignment 2 is disregarded. As shown in the lower part in figure 7, two generalizations are filtered for these alignments such that *(Peter)* $\leftrightarrow$ *(Peter)* and *(das Rad)* $\leftrightarrow$ *(the wheel)* are reduced into two reductions and *(Peter der Wolf)* $\leftrightarrow$ *(Peter the wolf)* is reduced into one reduction.

Hence, while correct brackets in one language might sometimes be difficult to determine, the brackets of the translation can be helpful to find their mutual analogous interpretation.

Two similar cases are shown in alignments 4 and 5. Here, the brackets *(Peter in das Haus)* and *(Peter ein Lügner)* are wrong. The algorithm overrules the wrong German brackets since their English translations are discontinuous. Instead most compositional, structural analogous generalizations are filtered by taking into consideration the brackets of the set of alignments as a whole and their sequences of aligned lexemes.

11 Conclusion

In this paper, an algorithm for sub-sentential alignment of aligned texts is discussed. The algorithm assumes both language sides to be linguistically bracketed. From the brackets, hypotheses of lexical transfer rules are extracted and a set of generalizations is generated. Each of these transfer rules is assigned a probability and a weight. From these rules, the algorithm, filters an invertible, structural analogous translation grammar. Implications and potentials of the approach are examined on a general level and a few examples applied to German-English alignments are presented in more detail.

References

Hans Ulrich Block. 2000. Example-Based Incremental Synchronous Interpretation. In *(Wahlster (ed.), 2000)*, pages 411–417.

Henrik Boström. 1999. Induction of Recursive Transfer Rules. In *Learning Language in Logic (LLL) Workshop*, Bled, Slovenia.

Ralph Brown. 2001. Transfer-Rule Induction for Example-Based Translation. In *Machine Translation Workshop of MT-Summit VIII.*

Michael Carl and Antje Schmidt-Wigger. 1998. Shallow Postmorphological Processing with KURD. In *Proceedings of NeMLaP3/CoNLL98*, pages 257–265, Sydney.

Michael Carl. 2001. Inducing probabilistic invertible translation grammars from aligned texts. In *CoNLL 2001.*

Halil Altay Güvenir and Ilyas Cicekli. 1998. Learning Translation Templates from Examples. *Information Systems*, 23(6):353–363.

Kevin McTait. 2001. Linguistic Knowledge and Complexity in an EBMT System Based on Translation Examples. In *Machine Translation Workshop of MT-Summit VIII.*

Dan I. Melamed. 2001. *Empirical Methods for Exploiting Parallel Texts.* MIT Press, Cambridge, MA.

Harold Somers. 1999. Review Article: Example-based Machine Translation. *Machine Translation*, 14(2):113–157.

Wolfgang Wahlster (ed.). 2000. *Verbmobil: Foundations of Speech-to-Speech Translation.* Springer, Heidelberg.

Menno van Zaanen. 2000. ABL: Alignment-Based Learning. In *COLING-2000*, pages 961–967.