A Systematic Evaluation of MBOT in Statistical Machine Translation

Nina Seemann seemanna@ims.uni-stuttgart.de
Fabienne Braune braunefe@ims.uni-stuttgart.de
Andreas Maletti maletti@ims.uni-stuttgart.de
Institute for Natural Language Processing, University of Stuttgart,
Pfaffenwaldring 5b, 70569 Stuttgart, Germany

Abstract

Shallow local multi-bottom up tree transducers (MBOTs) have been successfully used as translation models in several settings because of their ability to model discontinuities. In this contribution, several additional settings are explored and evaluated. The first rule extractions for tree-to-tree MBOT with non-minimal rules and for string-to-string MBOT are developed. All existing MBOT systems are systematically evaluated and compared to corresponding baseline systems in three large-scale translation tasks: English-to-German, English-to-Chinese, and English-to-Arabic. Particular emphasis is placed on the use of discontinuous rules. The developed rule extractions and analysis tools will be made publicly available.

1 Introduction

The area of statistical machine translation (SMT) (Koehn, 2009) concerns itself mostly with the development of automatic methods of deriving probabilistic translation models from a parallel corpus. Such a corpus contains corresponding sentences in two languages. The rule extraction mechanism automatically extracts rules for the translation model from such a corpus and assigns them a probability. Several different translation approaches are currently discussed in SMT. Among the best-performing systems one often finds phrase-based (Koehn et al., 2003) and syntax-based translation systems. Phrase-based systems use no linguistic information and can be obtained directly from the corpus. The same observation is true for hierarchical phrase-based systems (Chiang, 2005) as those are syntactic in a formal sense only. Syntax-based systems require some form of syntactic annotation, which is often represented as a (parse) tree. Correspondingly, we obtain several models, namely tree-to-tree, string-to-tree, and tree-to-string systems, which use trees on the source or the target language side. A multitude of formalisms has been proposed as syntax-based translation models. The most prominent might be the synchronous tree substitution grammars (STSGs) of Eisner (2003) and the non-contiguous synchronous tree sequence substituition grammars (STSSGs) of Sun et al. (2009). Recently, Maletti (2011) proposed *local multi bottom-up tree transducers* (MBOT) as a translation model for syntax-based SMT. An MBOT is an extension of an STSG that allows sequences of tree fragments on the target side of its rules. In this manner, it can model discontinuities. It can also be understood as a restricted form of a STSSG, in which the rules consist of sequences of source and target tree fragments.

Recently, MBOTs have been implemented as a translation model inside the Moses framework (Koehn et al., 2007) by Braune et al. (2013). Initially, only the rule extraction for minimal

tree-to-tree rules of Maletti (2011) was available. In combination with synchronous context-free grammar (SCFG) rules, already this system led to significant improvements over a corresponding system using SCFG rules only. Later, Seemann et al. (2015) proposed a rule extraction procedure for string-to-tree MBOTs that is able to extract non-minimal rules. Since the number of such rules usually explodes, certain (parametric) restrictions are imposed on the extracted rules in the same spirit as in (Chiang, 2005). Also in this setting, significant improvements over a corresponding SCFG-based system are obtained. However, systems imposing even further syntactic restriction still outperform those systems in some cases. To understand the benefits of the MBOT model and its dependence on syntax, we develop additional rule extractions for the missing scenarios and provide a systematic evaluation of MBOT-based systems. In particular, we develop a rule extraction for non-minimal tree-to-tree rules. It is known that minimal rules are rather restrictive (Galley et al., 2004), so we expect sizable improvements from the obtained tree-to-tree system (when compared to the tree-to-tree system using only minimal rules). In addition, we explore whether the discontinuous rules of MBOTs remain useful without any syntax. Consequently, we also develop a string-to-string rule extraction for MBOT. All the mentioned MBOT models are systematically evaluated and compared to popular translation models.

We evaluate our models on three large-scale translation tasks: English-to-German, English-to-Arabic, and English-to-Chinese. We demonstrate that the expected improvements from the non-minimal rules is indeed realized. However, the tree-to-tree SCFG baseline system, which also utilizes non-minimal rules, still outperforms the MBOT model. In the string-to-tree setting, MBOT significantly outperform SCFG as demonstrated by Seemann et al. (2015). Finally, in the string-to-string setting, discontinuous rules seem to be hardly useful at all. The obtained evaluation scores for such string-to-string (hierarchical) systems are comparable. Only a detailed analysis of the number of the used rules that are (potentially) discontinuous indeed confirms that hardly any discontinuous rules are used when decoding the test set. Overall, these results seem to suggest that discontinuous rules are most successful in the string-to-tree setting, where the strict syntactic structure of the output tree makes discontinuities valuable, while the flat structure of the input (string) allows sufficient freedom. Chiang (2010) arrives at a similar conclusion for general syntax-based systems with the argument that the target-side syntax might enable more grammatical translations. String-to-tree MBOTs offer an even better performance than string-to-tree SCFGs in our translation tasks, so discontinuities seem to be very relevant.

2 Statistical Machine Translation with MBOTs

Syntax-based statistical machine translation models use the syntactic structure of the input and/or output sentences during training and decoding. The syntactic structure of the sentences is typically automatically obtained with the help of a (constituent) parser, so we use 'parse' to refer to the syntactic structure. Several different settings can be distinguished depending on the use of parses:

- *tree-to-tree:* In this setting, parses are used for both the source and target sentences during training and for the input sentences during decoding.
- *tree-to-string:* Parses are used only for the source language; i.e., for the source sentences during training and for the input sentences during decoding.
- *string-to-tree:* In this setup, parses are only used for the target language sentences.
- *string-to-string:* No parses are used and the system is not syntax-based.

Shallow local multi bottom-up tree tranducers (MBOTs) have already been successfully applied as translation model in the tree-to-tree setting (Braune et al., 2013) as well as in the string-to-tree setting (Seemann et al., 2015). It is generally accepted that tree-to-string systems yield worse performance than string-to-tree systems (Chiang, 2010). We complete the picture by adding non-minimal rules to the tree-to-tree system, establishing a string-to-string variant (i.e., similar

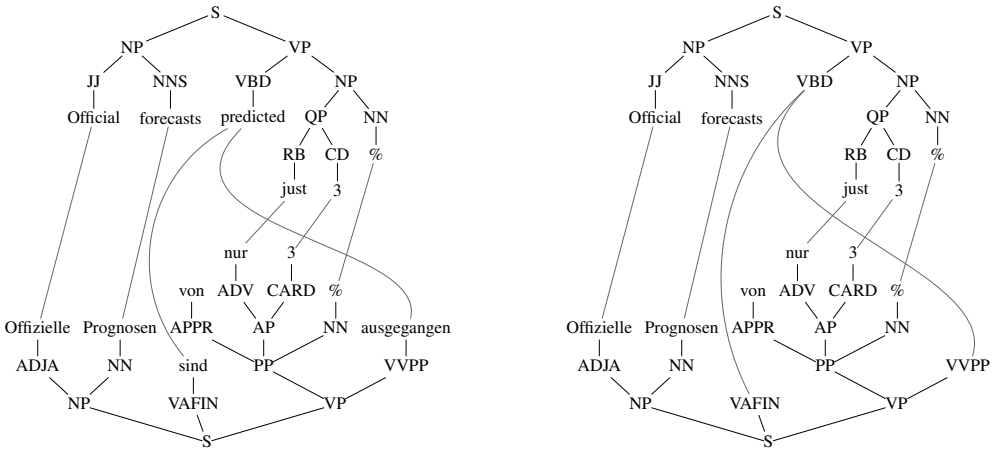

Figure 1: Word-aligned, bi-parsed sentence pair before (left) and after (right) excision of a rule.

to a hierarchical phrase-based model) that requires no parses at all, and providing an extensive evaluation of all those variants.

Let us start by illustrating the different variants. To this end, we first show the general rule shape of MBOTs, and then recall the existing rule extraction algorithms together with the particularities of the obtained rules. Our presentation is necessarily illustrative only. For the formal details we refer the reader to (Maletti, 2011; Braune et al., 2013; Seemann et al., 2015).

Roughly speaking, each MBOT has terminal symbols (e.g., lexical items) and nonterminal symbols (e.g., part-of-speech tags and syntactic categories). In essence, an MBOT is simply a finite set of rules. Each rule $\ell \to r$ consists of a left-hand side ℓ and a right-hand side r. The left-hand side $\ell = t$ consists of an object t (string or tree) for the source side, and the right-hand side $r = (t_1, \ldots, t_m)$ similarly consists of a *sequence* $t_1, \ldots, t_m$ of objects for the target side. The objects $t, t_1, \ldots, t_m$ are either strings or trees (with the restriction that $t_1, \ldots, t_m$ have the same type) formed from the terminal and nonterminal symbols with the additional restriction that each *exposed* occurrence of a nonterminal in an object in the right-hand side is linked to an *exposed* occurrence of a nonterminal in the left-hand side. More precisely, in a string, the lexical items are the terminal symbols and each nonterminal occurrence is exposed. In a tree, the lexical items only occur as leaves and additionally nonterminals are allowed as leaves. However, an occurrence of a nonterminal is exposed if and only if it is a leaf. The 4 different settings naturally correspond to the choice of strings or trees for the source and target side objects t and $t_1, \ldots, t_m$, respectively.

2.1 Minimal tree-to-tree rule extraction

Let us start with the tree-to-tree setting, for which a rule extraction for minimal rules[1] was proposed in (Maletti, 2011). In this case, the rule extraction requires a word-aligned, bi-parsed (parsed on both sides) parallel corpus. A sample entry of such a corpus is shown left in Figure 1. The rule extraction is applied to each sentence pair of the corpus and essentially performs the following steps:

(1) Select a minimal number of alignment edges E such that
   - the maximal (non-leaf nonterminal) source node v containing (as leaves) all sources of the selected edges and no sources of non-selected edges exists and

[1] A rule is minimal if it cannot be obtained by means of substitution from others.

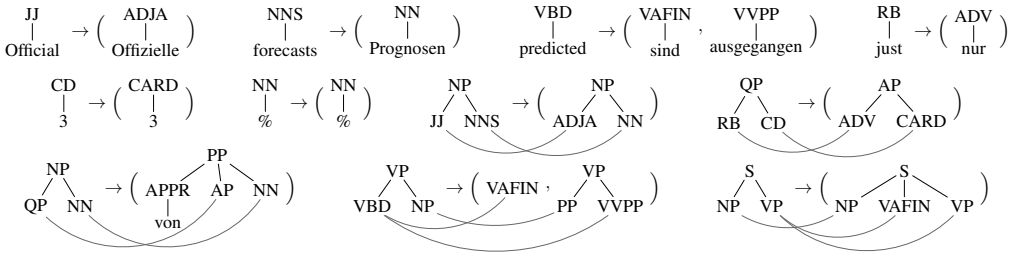

Figure 2: All minimal MBOT rules extractable from Figure 1.

- the maximal (non-leaf nonterminal) target nodes $w_1, \ldots, w_m$ containing all targets of the selected edges and no targets of non-selected edges exists.

In other words, the selected set E of edges admits maximal subtrees (below nodes v and $w_1, \ldots, w_m$) that are consistent with the word alignment (Chiang, 2005).

(2) Excise the subtree t below v for the source side and the subtrees $t_1, \ldots, t_m$ below $w_1, \ldots, w_m$, respectively, for the target side. In this way, we obtain the tree-to-tree rule $t \to (t_1, \ldots, t_m)$. After the excision, the nonterminals at v and $w_1, \ldots, w_m$ remain as leaf nonterminals and are linked. The result of excising the rule containing 'predicted' from the left entry of Figure 1 is shown right in Figure 1.

(3) Repeat the process with the linked pair of trees obtained after the excision.

Figure 2 shows all minimal MBOT rules that can be extracted from the word-aligned, bi-parsed sentence pair of Figure 1. The obtained rules are made shallow (Braune et al., 2013) by removing the internal (i.e., non-root and non-leaf) nodes. Figure 3 shows this process on the only rule of Figure 2 that is not yet shallow.

2.2 Non-minimal string-to-tree rule extraction

The parametrized non-minimal rule extraction of Seemann et al. (2015) for string-to-tree MBOT rules is an extension of the rule extraction of Chiang (2005) for hierarchical rules. As expected from the string-to-tree setting, it extracts rules from a word-aligned parallel corpus with parses for the target language side. Figure 4 shows an entry in such a corpus. Recall that a string-to-tree rule has the shape $w \to (t_1, \ldots, t_m)$ for a string w and trees $t_1, \ldots, t_m$. In the source string w, we only allow lexical items and the nonterminal X. The rule extraction proceeds similarly as described in Section 2.1 with the exception that a phrase (or span) is selected in the source side (instead of a node of the tree) and that the minimality and maximality conditions are dropped (non-minimality). When excising from the source side, we leave the nonterminal X instead of the excised material. In essence, we obtain *all* string-to-tree rules that are consistently word-aligned in this manner. However, there are way too many such rules, and Seemann et al. (2015) establish the following conditions on w that need to be fulfilled for a rule to be extracted.

- It should have (i) a lexical item that is the source of a word alignment or (ii) an occurrence of X (i.e., selecting the empty set E of edges is excluded).
- It should correspond to a span of length at most 10 and contain at most 5 occurrences of

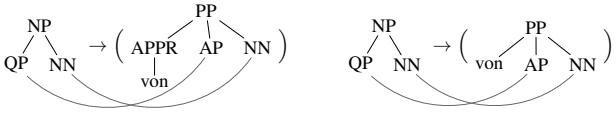

Figure 3: Non-shallow MBOT rule (left) and its shallow counterpart (right).

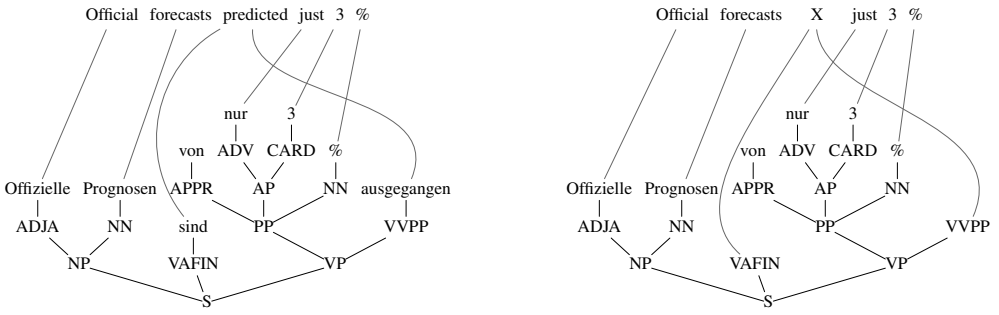

Figure 4: Sentence pair with target side parse before (left) and after (right) excision of a rule.

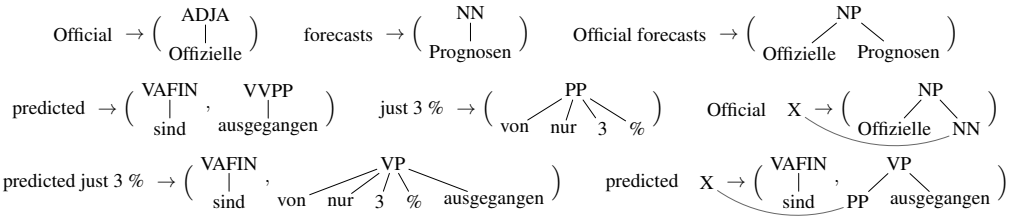

Figure 5: String-to-tree rules extracted from the sentence pair of Figure 4.

lexical items or X.

- It cannot start with X and consecutive 'X' are forbidden.

All extracted rules are made shallow as in Section 2.1. In Figure 5 we illustrate some string-to-tree rules that are extracted from the sentence pair left in Figure 4. The excision of the rule with left-hand side 'predicted' from the sentence pair is illustrated right in Figure 4.

3 Non-minimal tree-to-tree rule extraction

To provide a complete picture, we also want to consider a non-minimal tree-to-tree rule extraction. To this end, we modify the (non-minimal) string-to-tree rule extraction of Section 2.2 to extract tree-to-tree rules as follows. We first extract string-to-tree rules, so let $r = w \to (t_1, \ldots, t_m)$ be such an extracted rule. Let $w = w_1 \cdots w_n$ be the decomposition of the string w into tokens. Since we want to extract tree-to-tree rules, our sentence pairs are now bi-parsed (as in Figure 1), so a parse of the source sentence is available. Based on this parse, we determine whether the left-hand side w corresponds to a constituent in it. If it corresponds to a constituent labeled N, then we construct the tree-to-tree rule $N(w'_1, \ldots, w'_n) \to (t_1, \ldots, t_m)$, where w'_i is simply w_i for all lexical items w_i and the corresponding constituent for $w_i = \text{X}$. Otherwise we ignore the string-to-tree rule r and proceed with the next one. For example, the string-to-tree rule

$$\text{predicted just} \to \big(\text{VAFIN}(\text{sind}),\ \text{ADV}(\text{nur}),\ \text{VVPP}(\text{ausgegangen})\big)$$

is extracted in the string-to-tree setting for the sentence pair of Figure 4 (left), but 'predicted just' is not a constituent in the source sentence parse of Figure 1 (left). Consequently, this string-to-tree rule is ignored.

Since the left hand side w has to match a constituent, the number of extractable rules is drastically lower than in the string-to-tree setting. Hence, we can remove the last of the conditions described above. Note that rules like those in Figure 2 can be extracted using the

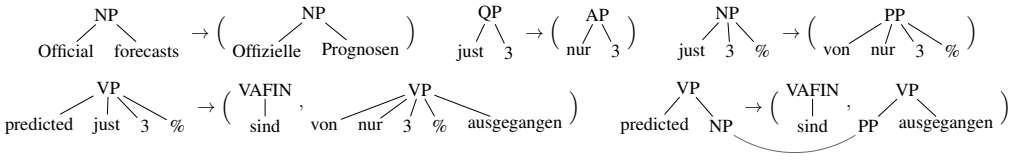

Figure 6: Tree-to-tree rules extracted by non-minimal rule extraction.

non-minimal tree-to-tree rule extraction. Some additional tree-to-tree rules that can be extracted from the word-aligned, bi-parsed sentence pair left in Figure 1 are displayed in Figure 6.

4 String-to-string rule extraction

Secondly, we want to completely abandon the syntactic annotation and derive a rule extraction for string-to-string MBOT rules. We again achieve this by a simple modification of the existing string-to-tree rule extraction of Seemann et al. (2015). Overall, string-to-string MBOT rules are similar to hierachical rules (Chiang, 2005), but with a sequence of strings in the right-hand side. We now have no parses at all, so our training data is a simple word-aligned parallel corpus. An entry of such a corpus is displayed in Figure 7. To accommodate this situation, we perform the same changes mentioned in the step from the tree-to-tree rule extraction to the string-to-tree rule extraction also for the target side. Thus, we no longer need to identify a node in the target sentence parse, but rather identify a sequence of phrases (or spans) that are consistent with the word alignment. The additional restrictions on the left-hand side w in the string-to-tree rule extraction are also imposed onto the left-hand sides of the extracted string-to-string rules, but we do not impose them onto the right-hand side. These restrictions were imposed to reasonably limit the number of rules, but it shows that restricting the right-hand side is (generally) not necessary.

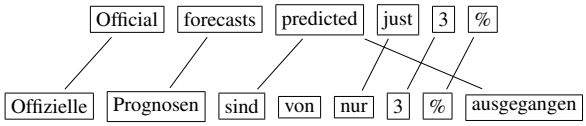

Figure 7: Word-aligned sentence pair.

From the sentence pair in Figure 7 we can, for example, extract the following rules. We indicate linked nonterminals by using the same subscript on them.

- Official $\rightarrow$ (Offizielle)
- forecasts $\rightarrow$ (Prognosen)
- predicted just $\rightarrow$ (sind , nur , ausgegangen)
- predicted just $\rightarrow$ (sind von nur , ausgegangen)
- just 3 % $\rightarrow$ (nur 3 %)
- just 3 % $\rightarrow$ (von nur 3 %)
- predicted just 3 % $\rightarrow$ (sind , nur 3 % ausgegangen)
- predicted just 3 % $\rightarrow$ (sind von nur 3 % ausgegangen)
- Official X_1 $\rightarrow$ (Offizielle X_1)
- predicted X_1 $\rightarrow$ (sind , nur , X_1 ausgegangen)

5 Experimental Evaluation

Our main contribution is the experimental evaluation of MBOTs in the various settings (tree-to-tree, string-to-tree, and string-to-string). We also compare to standard models (SCFG and

	English to German	English to Arabic	English to Chinese
training data	7th EuroParl (Koehn, 2005)	MultiUN (Eisele and Chen, 2010)	
training data size	≈ 1.8M sentence pairs	≈ 5.7M sentence pairs	≈ 1.9M sentence pairs
language model	5-gram SRILM (Stolcke, 2002)		
add. LM data	WMT 2013	Arabic in MultiUN	Chinese in MultiUN
LM data size	≈ 57M sentences	≈ 9.7M sentences	≈ 9.5M sentences
tuning data	WMT 2013	cut from MultiUN	NIST 2002, 2003, 2005
tuning size	3,000 sentences	2,000 sentences	2,879 sentences
test data	WMT 2013 (Bojar et al., 2013)	cut from MultiUN	NIST 2008 (NIST, 2010)
test size	3,000 sentences	1,000 sentences	1,859 sentences

Table 1: Summary of the used resources.

hierarchical models) in order to evaluate the effect of the discontinuities offered by the MBOT model. We try to be comprehensive, but naturally we can only report results for a limited number of experiments. We chose to perform experiments for the translation directions English-to-German, English-to-Arabic, and English-to-Chinese. The languages were selected such that constituency parsers and large parallel corpora are readily available. In addition, we selected target languages, in which the discontinuity offered by the MBOT model might be useful.

5.1 Resources

For better comparisons, we use exactly the same resources as Seemann et al. (2015) for the evaluation. We summarize the experimental setup in Table 1. We applied length-ratio filtering to all data sets. Furthermore, all training sets have been word aligned using GIZA++ (Och and Ney, 2003) using the *grow-diag-final-and* heuristic (Koehn et al., 2005).

The tasks required various forms of preprocessing of the data. The English (source) side of the training data was true-cased and parsed with the provided grammar of the Berkeley parser (Petrov et al., 2006). Next, we comment on the preprocessing tasks that are specific for each translation task.

- *English-to-German:* The German text was also true-cased and parsed with the provided grammar of BitPar (Schmid, 2004). German is morphologically rich and in order to avoid sparseness issues, we removed the functional and morphological annotation from the tags used in the parses.
- *English-to-Arabic:* The Arabic text was tokenized with MADA (Habash et al., 2009) and transliterated according to Buckwalter (2002). Since the Berkeley parser (Petrov et al., 2006) also provides a grammar for Arabic, we parsed the Arabic training data with it.
- *English-to-Chinese:* The Chinese sentences were word-segmented using the Stanford Word Segmenter (Chang et al., 2008). Again, the Berkeley parser (Petrov et al., 2006) with its provided grammar delivers the parse trees for the Chinese training data.

After the preprocessing steps, we obtained a word-aligned, bi-parsed parallel corpus, to which we applied the described rule extractions together with the baseline rule extractions provided by Moses (Koehn et al., 2007; Hoang et al., 2009). To give a quick overview, we report the number of extracted rules for all translation tasks and rule extractions in Table 2. We can immediately confirm that relaxing the conditions during rule extraction (e.g., from tree-to-tree to string-to-tree) greatly increases the number of extracted rules for each translation task. For example, the string-to-tree rule extraction for MBOTs (Seemann et al., 2015) extracts 12–17 times more rules than the minimal tree-to-tree rule extraction for MBOTs (Maletti, 2011). In addition, the availability of several target-side objects (and thus discontinuities) also leads to additional freedom during rule extraction, which is evidenced by the larger numbers of rules extracted for MBOTs compared to those extracted for the SCFG baseline. For our experiments we heavily

System	Number of extracted rules		
	English-To-German	English-To-Arabic	English-To-Chinese
tree-to-tree SCFG (baseline)	6,630,590	24,358,001	8,161,362
minimal tree-to-tree MBOT	12,478,160	28,725,229	10,162,325
tree-to-tree MBOT	40,736,687	151,322,970	84,220,528
string-to-tree SCFG (baseline)	14,092,729	55,169,043	17,047,570
string-to-tree MBOT	143,661,376	491,307,787	162,240,663
hierarchical SCFG (baseline)	406,433,344	842,290,537	209,482,192
string-to-string MBOT	1,084,007,782	2,208,445,501	682,505,767

Table 2: Number of extracted rules for the different rule extractions.

parallelized the processes using different architectures. Unfortunately, this precludes a sensible analysis of decoding times.

5.2 Translation features

As usual, the task of the decoder is to find the best translation $\hat{f}$ of the input object e (string or tree) licensed by the translation model and the language model.

$$\hat{f} = \arg\max_f p(f \mid e) = \arg\max_f p(e \mid f) \cdot p(f)$$

The probability $p(f)$ is provided by a (string) 5-gram language model for the target language. Thus, if target syntax is used, then the yield (the sentence written on the frontier) of the tree f is used for language model scoring. The data used to train the language model and the used toolkit are reported in Table 1. The translation model provides the probability $p(e \mid f)$ and uses either the MBOT model or the baseline SCFG model as implemented in Moses. More precisely, the translation model[2] uses a log-linear model (Och, 2003) of weighted features $h_k(\cdot)^{\lambda_k}$ over derivations D for the pair (e, f).

$$p(e \mid f) = \max_{D \text{ derivation for } (e,\, f)} p(D) = \max_{D \text{ derivation for } (e,\, f)} \Big(\prod_i h_i(D)^{\lambda_i} \Big) \ ,$$

where $h_i(\cdot)$ are features on derivations. The features of the derivation are usually derived as a product of the rule features of those rules that constitute the derivation. We used the following (mostly standard) rule features (Koehn, 2009):

- the forward and backward translation probabilities,
- the forward and backward lexical translation probabilities,
- the phrase and word penalty, and
- the *gap penalty*, which is specific for MBOTs.

The forward and backward translation probabilities are obtained as normalized relative frequencies. We applied Good-Turing smoothing (Good, 1953) to all rules that were extracted at most 10 times. Both lexical translation probabilities are obtained as usual, and the MBOT-specific gap penalty is defined as 100^{1-c}, where c is the number of target objects used in all rules that contributed to D. This feature is intended to allow the model to tune the amount of discontinuity to the specific target language. Indeed in all experiments the feature weights λ_i of the log-linear model were trained using minimum error rate training (Och, 2003).

The task of the decoder is the identification of the best-scoring target object f and derivation D in the above definition of $p(e \mid f)$. In all our experiments a CYK+ chart parser is used as decoder. The decoder for the SCFG model is provided by the syntax component (Hoang et al., 2009) of the Moses framework, and the decoder for the MBOT model is provided by MbotMoses branch (Braune et al., 2013) of Moses.

[2]Indeed the language model is also scaled with a feature weight λ.

		BLEU		
Setting	**System**	**En-De**	**En-Ar**	**En-Zh**
tree-to-tree	Moses (baseline)	14.50	43.49	17.63
	minimal MBOT	14.09	32.88	12.01
	non-minimal MBOT	14.41	41.37	16.77
string-to-tree	Moses (baseline)	14.96	48.23	17.69
	MBOT	*15.49	*49.10	*18.35
	GHKM (Galley et al., 2004, 2006)	17.10	46.66	18.33
hierarchical	Moses (baseline)	17.00	51.71	18.74
	MBOT	16.57	—	18.60
phrase-based Moses		16.80	51.90	18.09

Table 3: BLEU evaluation results for all 3 translation tasks. Starred results indicate statistically significant improvements over the baseline (at confidence $p < 1\%$).

5.3 Quantitative evaluation

In this section, we first compare all systems to each other using the score BLEU (Papineni et al., 2002). We also present the results obtained by systems that were high-ranked on public shared tasks (Bojar et al., 2014) such as phrase-based systems (Koehn et al., 2003) or string-to-tree systems obtained following Galley et al. (2004, 2006). All systems were tuned for BLEU on the tuning data, and we report the BLEU scores obtained by the tuned systems on the test sets. The MBOT-based systems were evaluated against their corresponding syntax component (Hoang et al., 2009) of the Moses toolkit, which implements tree-to-tree, string-to-tree, and string-to-string (hierarchical) rule extractions. All of them follow essentially the procedure outlined in Chiang (2005), which was also the basis for the rule extraction of Seemann et al. (2015) and our string-to-string rule extraction. Our implementation of the non-minimal tree-to-tree MBOT rule extraction is also an extension of the corresponding procedure of the syntax component of Moses. We also checked statistical significance for the MBOT results using the implementation of Gimpel (2011).

We performed large scale experiments on three major translation tasks, namely English-to-German (En-De), English-to-Arabic (En-Ar), and English-to-Chinese (En-Zh). The goal was to evaluate the following MBOT systems: (i) the minimal tree-to-tree system (Section 2.1), (ii) the non-minimal tree-to-tree system (Section 3), (iii) the non-minimal string-to-tree system (Section 2.2), and (iv) the string-to-string system (Section 4). The obtained results are reported in Table 3. Unfortunately, the rule table of the string-to-string MBOT for the English-to-Arabic translation task — although already filtered on the given input — was too large to load into main memory (available: 500GB RAM).

Let us now discuss the results for the various settings. Overall, we observe that the tree-to-tree systems perform worst. For the baseline system using SCFG rules (i.e., MBOT rules with a single tree on both the left- and right-hand side), this result is not surprising. Already, Ambati and Lavie (2008) have shown that tree-to-tree rules are too restrictive to achieve good lexical coverage. However, our results show that making rules more flexible by allowing several target trees hurts the performance instead of yielding improvements. This effect is particularly visible when only using minimal rules. On the English-to-Arabic and English-to-Chinese translation tasks, the minimal MBOT system loses 10.61 and 5.62 BLEU points, respectively, over the baseline. Interestingly, on the English-to-German translation task the loss is only 0.41 BLEU points. Adding non-minimal MBOT rules yields the expected large improvements, but is overall still not good enough to beat the tree-to-tree baseline. This result is interesting insofar as it

English-to-German									
					Target tree fragments				
MBOT variant	**Type**	**Lex**	**Struct**	**Total**	2	3	4	5	≥ 6
minimal t-to-t	cont.	55,910	4,492	60,402					
	discont.	2,167	7,386	9,553	6,458	2,389	471	34	1
non-minimal t-to-t	cont.	44,951	2,850	47,801					
	discont.	4,149	2,348	6,497	5,601	821	62	13	–
non-minimal s-to-t	cont.	27,351	635	27,986					
	discont.	9,336	1,110	10,446	5,565	3,441	1,076	312	52
non-minimal s-to-s	cont.	29,972	3,600	33,572					
	discont.	259	131	390	387	3	–	–	–

Table 4: Number of rules per type used when decoding test (Lex = lexical rules; Struct = structural rules; [dis]cont. = [dis]contiguous).

does not confirm the results of Sun et al. (2009) on large scale experiments with target side discontinuities.[3]

The results for the string-to-tree setting are much better than those for the tree-to-tree systems (see Table 3). The BLEU score improvements are not very pronounced for English-to-German and English-to-Chinese, but on the English-to-Arabic translation task the string-to-tree systems (both baseline and MBOT) achieve huge improvements. Those systems come in at 4.74 and 5.61 BLEU points, respectively, ahead of the tree-to-tree baseline. As already demonstrated by Seemann et al. (2015), the MBOT system yields significant improvements over the baseline on all those language pairs. The GHKM systems achieve mixed results. They outperform the MBOT system on English-to-German, achieve the same performance as the MBOT system on English-to-Chinese, and lose even against the baseline on the English-to-Arabic translation task.

Finally, the string-to-string systems generally yield the best translation quality (as measured by BLEU). The experiments for the English-to-German and English-to-Chinese translation task show that our string-to-string MBOT system does not improve performance in these cases. Indeed, the analysis presented in Section 6 suggests that the string-to-string rules are flexible enough to achieve high coverage even without the need for multiple phrases in the right-hand side. This is slightly disappointing as Galley and Manning (2010) incorporated discontinuous phrases into a phrase-based system, and their evaluation on Chinese-to-English showed significant improvements over a standard phrase-based baseline as well as over a hierarchical baseline. However, the differences are generally not large, and even the phrase-based system achieves similar performance.

6 Analysis of Discontinuity

Another goal was to identify whether discontinuous rules are useful and to what extent these are useful. We try to estimate their impact on the translation quality by inspecting the statistics on the rules used in the derivations. Consequently, only rules that produce part of the final output in each of the translation tasks count. The current tools of MbotMoses (Braune et al., 2013) only allow the counting of rules used during decoding. At present, it is infeasible to track discontinuous objects through the derivation to decide whether they are actually assembled continuously or discontinuously. Thus, discontinuous rules only indicate a potential discontinuity.

[3] The experiments of Sun et al. (2009) report scores for the translation task Chinese-to-English for systems trained on 240,000 sentences only. Their model allows discontinuities on the source language side, which should be comparable to target-side discontinuities for the opposite translation direction English-to-Chinese.

English-to-Arabic									
					Target tree fragments				
MBOT variant	**Type**	**Lex**	**Struct**	**Total**	2	3	4	5	≥ 6
minimal t-to-t	cont.	18,389	2,855	21,244					
	discont.	1,138	1,920	3,085	2,525	455	67	8	3
non-minimal t-to-t	cont.	9,826	1,581	11,407					
	discont.	1,605	746	2,315	1,577	565	158	38	13
non-minimal s-to-t	cont.	1,839	651	2,490					
	discont.	3,670	1,324	4,994	3,008	1,269	528	153	36

Table 5: Number of rules per type used when decoding test (Lex = lexical rules; Struct = structural rules; [dis]cont. = [dis]contiguous).

Tables 4, 5, and 6 show the statistics on the rules used during decoding. Continuous rules (i.e., rules with a single object in the right-hand side) are abbreviated by *cont*, and (potentially) discontinuous rules are abbreviated by *discont*. To provide a deeper analysis, we also distinguish between lexical and structural rules. Lexical rules, abbreviated *Lex*, are rules that contain no exposed nonterminal symbols. Similarly structural rules, abbreviated *Struct*, are rules containing at least one such nonterminal symbol. Finally, we abbreviate the settings tree-to-tree, string-to-tree, and string-to-string by 't-to-t', 's-to-t', and 's-to-s', respectively.

We first discuss the results for the tree-to-tree systems presented across Tables 4, 5, and 6. If we only use minimal MBOT rules, then 11% of the rules used during decoding are discontiguous in the English-to-German and English-to-Chinese translation tasks. The rate is slightly higher in the English-to-Arabic translation task (14.5%). For all the translation tasks, the majority of the discontinuous rules are structural. This fact is not very surprising since the leaves of the minimal tree-to-tree rules are either lexical items or exposed non-terminal occurrences. The minimality constraint encourages word-by-word translation, and once the lexical rules are excised, only structural rules remain. Based on the observed high BLEU score losses, it seems that minimal tree-to-tree rules are, at present, unable to correctly assemble discontinuous parts.

If we additionally use non-minimal tree-to-tree rules, then the rates of discontinuous rules change. For English-to-German the rate remains almost the same at 13%, whereas for the English-to-Arabic translation task it increases to 20%. Finally, for English-to-Chinese suddenly only 4% of the rules applied during decoding are discontinuous. The non-minimality encourages large rules, which are more likely to contain only lexical items. As expected, the number of discontinuous lexical rules is always larger than the number of discontinuous structural rules in this setting. This is particularly true for English-to-German and English-to-Arabic, where two third of the discontinuous rules are lexical, whereas the distribution is almost even for English-to-Chinese. We believe that these lexical discontinuous rules capture relevant idiomatic expressions or encode agreement or correspondences, which yield the large improvements in translation quality over minimal rules only.

For string-to-tree systems we only present the statistics since they have been discussed already by Seemann et al. (2015). It is evident that they generally use the largest amounts of discontinuous rules, which is rewarded with significant improvements over the baseline system without discontinuieties.

Finally, for the string-to-string systems, the opposite situation presents itself. Here, the number of discontinuous rules is indeed marginal. On the English-to-German translation task only 1.1% of 33,962 rules are discontinuous. The English-to-Chinese system also only uses 2.3% discontinuous rules (out of 25,575 rules). We believe that the low use of discontinuous string-to-string rules can be explained by the absence of linguistic annotations. Without them,

English-to-Chinese									
					Target tree fragments				
MBOT variant	**Type**	**Lex**	**Struct**	**Total**	2	3	4	5	≥ 6
minimal t-to-t	cont.	34,275	8,820	43,095					
	discont.	516	4,292	4,808	3,816	900	82	6	4
non-minimal t-to-t	cont.	35,031	2,045	37,076					
	discont.	771	744	1,515	1,222	248	36	4	5
non-minimal s-to-t	cont.	17,135	1,585	18,720					
	discont.	4,822	3,341	8,163	6,411	1,448	247	55	2
non-minimal s-to-s	cont.	15,769	9,208	24,977					
	discont.	108	490	598	591	7	–	–	–

Table 6: Number of rules per type used when decoding test (Lex = lexical rules; Struct = structural rules; [dis]cont. = [dis]contiguous).

the rules become very flexible, thus removing the need for discontinuous MBOT rules in this setting. Since the number of used discontinuous rules is so low, it can be assumed that essentially the same rules were used during decoding when comparing the MBOT system to the baseline. This would also explain their comparable BLEU scores.

7 Conclusion

We have extended the existing rule extraction techniques for shallow local multi bottom-up tree transducers to the two main missing settings. First, we designed a non-minimal tree-to-tree rule extraction for MBOT, which extends the corresponding rule extraction for minimal rules. Secondly, we developed a rule extraction for the string-to-string setting, which does not rely on syntactical information. Naturally, we also evaluated these new rule extractions together with several other systems in 3 large scale translation tasks (English-to-German, English-to-Arabic, and English-to-Chinese).

As expected, the non-minimal tree-to-tree system performs much better than the corresponding system using only minimal rules, but even the system with non-minimal rules does not beat the SCFG baseline (using non-minimal rules). It seems that discontinuity remains a challenge for tree-to-tree rules. Overall, tree-to-tree systems report the worst scores. For the string-to-tree systems already Seemann et al. (2015) report significant improvements in translation quality when using discontinuous rules. Finally, in the string-to-string (hierarchical) setting, discontinuous rules are hardly ever used, so when compared to the SCFG baseline essentially the same performance is obtained. Most likely, hierarchical rules are flexible enough to handle most common forms of discontinuity without the need to explicitly represent it in its rules. In summary, since MBOT offers certain consistent advantages across the different language pairs it may be useful to exploit a hybrid approach in the future.

To support further experimentation by the community, we publicly release our developed software and analysis tools (`http://www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/mbotmoses.en.html`).

Acknowledgement

The authors would like to thank the reviewers for their helpful comments and suggestions.

All authors were financially supported by the German Research Foundation (DFG) grant MA 4959 / 1-1, which we gratefully acknowledge.

References

Ambati, V. and Lavie, A. (2008). Improving syntax driven translation models by re-structuring divergent and non-isomorphic parse tree structures. In *Proc. AMTA 2008*, pages 235–244.

Bojar, O., Buck, C., Callison-Burch, C., Federmann, C., Haddow, B., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2013). Findings of the 2013 Workshop on Statistical Machine Translation. In *Proc. 8th WMT*, pages 1–44. Association for Computational Linguistics.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., Soricut, R., Specia, L., and Tamchyna, A. (2014). Findings of the 2014 workshop on statistical machine translation. In *Ninth Workshop on Statistical Machine Translation*, WMT, pages 12–58, Baltimore, Maryland.

Braune, F., Seemann, N., Quernheim, D., and Maletti, A. (2013). Shallow local multi bottom-up tree transducers in statistical machine translation. In *Proc. 51st Annual Meeting of the Association for Computational Linguistics*, pages 811–821.

Buckwalter, T. (2002). Arabic transliteration. `http://www.qamus.org/transliteration.htm`.

Chang, P.-C., Galley, M., and Manning, C. D. (2008). Optimizing Chinese word segmentation for machine translation performance. In *Proc. 3rd WMT*, pages 224–232. Association for Computational Linguistics.

Chiang, D. (2005). Hierarchical phrase-based translation. In *Proceedings of the 43th Annual Meeting of the Association of Computational Linguistics*, page 263270. Association for Computational Linguistics.

Chiang, D. (2010). Learning to translate with source and target syntax. In *Proc. 48th ACL*, pages 1443–1452. Association for Computational Linguistics.

Eisele, A. and Chen, Y. (2010). MultiUN: A multilingual corpus from United Nation documents. In *Proc. 7th LREC*, pages 2868–2872. European Language Resources Association.

Eisner, J. (2003). Learning non-isomorphic tree mappings for machine translation. In *Proceedings of the 41th Annual Meeting of the Association of Computational Linguistics*, pages 205–208.

Galley, M., Graehl, J., Knight, K., Marcu, D., Deneefe, S., Wang, W., and Thayer, I. (2006). Scalable inference and training of context-rich syntactic translation models. In *In ACL*, pages 961–968.

Galley, M., Hopkins, M., Knight, K., and Marcu, D. (2004). What's in a translation rule? In *HLT-NAACL*.

Galley, M. and Manning, C. D. (2010). Accurate non-hierarchical phrase-based translation. In *HLT-NAACL*, pages 966–974. The Association for Computational Linguistics.

Gimpel, K. (2011). Code for statistical significance testing for MT evaluation metrics. `http://www.ark.cs.cmu.edu/MT/`.

Good, I. J. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3–4):237–264.

Habash, N., Rambow, O., and Roth, R. (2009). MADA+TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In *Proc. 2nd MEDAR*, pages 102–109. Association for Computational Linguistics.

Hoang, H., Koehn, P., and Lopez, A. (2009). A unified framework for phrase-based, hierarchical, and syntax-based statistical machine translation. In *In Proceedings of the International Workshop on Spoken Language Translation (IWSLT*, pages 152–159.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proc. 10th MT Summit*, pages 79–86. Association for Machine Translation in the Americas.

Koehn, P. (2009). *Statistical Machine Translation*. Cambridge University Press.

Koehn, P., Axelrod, A., Mayne, A. B., Callison-Burch, C., Osborne, M., and Talbot, D. (2005). Edinburgh system description for the 2005 IWSLT Speech Translation Evaluation. In *Proc. 2nd IWSLT*, pages 68–75.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proc. of ACL: Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. ACL.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 48–54. Association for Computational Linguistics.

Maletti, A. (2011). How to train your multi bottom-up tree transducer. In *Proceedings of the 49th Annual Meeting of the Association of Computational Linguistics*, pages 825–834.

NIST (2010). NIST 2002 [2003, 2005, 2008] open machine translation evaluation. Linguistic Data Consortium. LDC2010T10 [T11, T14, T21].

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proc. 41st ACL*, pages 160–167. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Papineni, K., Roukos, S., Ward, T., and jing Zhu, W. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proc. 40th ACL*, pages 311–318. Association for Computational Linguistics.

Petrov, S., Barrett, L., Thibaux, R., and Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. In *Proc. 44th ACL*, pages 433–440. Association for Computational Linguistics.

Schmid, H. (2004). Efficient parsing of highly ambiguous context-free grammars with bit vectors. In *Proc. 20th COLING*, pages 162–168. Association for Computational Linguistics.

Seemann, N., Braune, F., and Maletti, A. (2015). String-to-tree multi bottom-up tree transducers. In *Proc. 53rd ACL*, pages 815–824. Association for Computational Linguistics.

Stolcke, A. (2002). SRILM — an extensible language modeling toolkit. In *Proc. 7th INTERSPEECH*, pages 257–286.

Sun, J., Zhang, M., and Tan, C. L. (2009). A non-contiguous tree sequence alignment-based model for statistical machine translation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL*, pages 914–922.