String-to-Tree Multi Bottom-up Tree Transducers

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

Abstract

We achieve significant improvements in several syntax-based machine translation experiments using a string-to-tree variant of multi bottom-up tree transducers. Our new parameterized rule extraction algorithm extracts string-to-tree rules that can be discontiguous and non-minimal in contrast to existing algorithms for the tree-to-tree setting. The obtained models significantly outperform the string-to-tree component of the Moses framework in a large-scale empirical evaluation on several known translation tasks. Our linguistic analysis reveals the remarkable benefits of discontiguous and non-minimal rules.

1 Introduction

We present an application of a variant of local multi bottom-up tree transducers (ℓMBOTs) as proposed in Maletti (2011) to statistical machine translation. ℓMBOTs allow discontinuities on the target language side since they have a sequence of target tree fragments instead of a single tree fragment in their rules. The original approach makes use of syntactic information on both the source and the target side (*tree-to-tree*) and a corresponding minimal rule extraction is presented in (Maletti, 2011). Braune et al. (2013) implemented it as well as a decoder inside the Moses framework (Koehn et al., 2007) and demonstrated that the resulting tree-to-tree ℓMBOT system significantly improved over its tree-to-tree baseline using minimal rules. We can see at least two drawbacks in this approach. First, experiments investigating the integration of syntactic information on both sides generally report quality deterioration. For example, Lavie et al. (2008), Liu et al. (2009), and Chiang (2010) noted that translation quality tends to decrease in tree-to-tree systems because the rules become too restrictive. Second, minimal rules (i.e., rules that cannot be obtained from other extracted rules) typically consist of a few lexical items only and are thus not the most suitable to translate idiomatic expressions and other fixed phrases. To overcome these drawbacks, we abolish the syntactic information for the source side and develop a string-to-tree variant of ℓMBOTs. In addition, we develop a new rule extraction algorithm that can also extract non-minimal rules. In general, the number of extractable rules explodes, so our rule extraction places parameterized restrictions on the extracted rules in the same spirit as in (Chiang, 2007). In this manner, we combine the advantages of the hierarchical phrase-based approach on the source side and the tree-based approach with discontinuiety on the target side.

We evaluate our new system in 3 large-scale experiments using translation tasks, in which we expect discontinuiety on the target. MBOTs are powerful but asymmetric models since discontinuiety is available only on the target. We chose to translate from English to German, Arabic, and Chinese. In all experiments our new system significantly outperforms the string-to-tree syntax-based component (Hoang et al., 2009) of Moses. The (potentially) discontiguous rules of our model are very useful in these setups, which we confirm in a quantitative and qualitative analysis.

2 Related work

Modern statistical machine translation systems (Koehn, 2009) are based on different translation models. Syntax-based systems have become widely used because of their ability to handle non-local reordering and other linguistic phenomena better than phrase-based models (Och and Ney, 2004). Synchronous tree substitution grammars (STSGs) of Eisner (2003) use a single source and target tree fragment per rule. In contrast, an ℓMBOT rule contains a single source tree

Proceedings of the 53th Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 815–824,
Beijing, China, July 26–31, 2015.

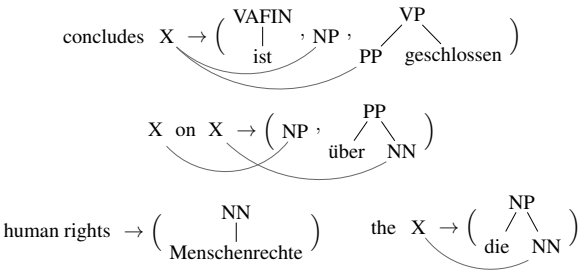

Figure 1: Several valid rules for our MBOT.

fragment and a sequence of target tree fragments. ℓMBOTs can also be understood as a restriction of the non-contiguous STSSGs of Sun et al. (2009), which allow a sequence of source tree fragments and a sequence of target tree fragments. ℓMBOT rules require exactly one source tree fragment.

While the mentioned syntax-based models use tree fragments for source and target (tree-to-tree), Galley et al. (2004) and Galley et al. (2006) use syntactic annotations only on the target language side (*string-to-tree*). Further research by DeNeefe et al. (2007) revealed that adding non-minimal rules improves translation quality in this setting. Here we improve statistical machine translation in this setting even further using non-minimal ℓMBOT rules.

3 Theoretical Model

As our translation model, we use a string-to-tree variant of the shallow local multi bottom-up tree transducer of Braune et al. (2013). We will call our variant MBOT for simplicity. Our MBOT is a synchronous grammar (Chiang, 2006) similar to a synchronous context-free grammar (SCFG), but instead of a single source and target fragment per rule, our rules are of the form $s \rightarrow (t_1, \ldots, t_n)$ with a single *source string* s and potentially several *target tree fragments* $t_1, \ldots, t_n$. Besides lexical items the source string can contain (several occurrences of) the placeholder X, which links to non-lexical leaves in the target tree fragments. In contrast to an SCFG each placeholder can have several such links. However, each non-lexical leaf in a target tree fragment has exactly one such link to a placeholder X. An MBOT is simply a finite collection of such rules. Several valid rules are depicted in Figure 1.

The sentential forms of our MBOTs, which occur during derivations, have exactly the same shape as our rules and each rule is a sentential

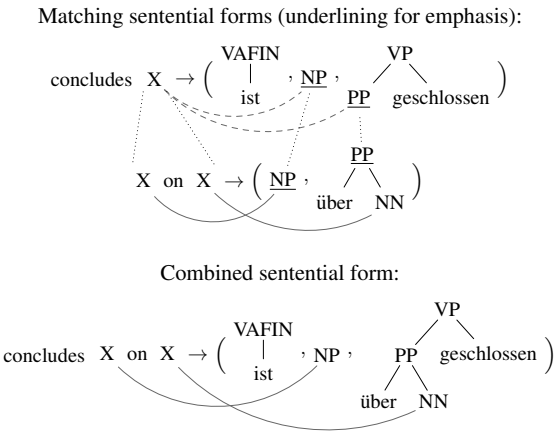

Figure 2: Substitution of sentential forms.

form. We can combine sentential forms with the help of substitution (Chiang, 2006). Roughly speaking, in a sentential form ξ we can replace a placeholder X that is linked (left-to-right) to non-lexical leaves $C_1, \ldots, C_k$ in the target tree fragments by the source string of any sentential form ζ, whose roots of the target tree fragments (left-to-right) read $C_1, \ldots, C_k$. The target tree fragments of ζ will replace the respective linked leaves in the target tree fragments of the sentential form ξ. In other words, substitution has to respect the symbols in the linked target tree fragments and all linked leaves are replaced at the same time. We illustrate substitution in Figure 2, where we replace the placeholder X in the source string, which is linked to the underlined leaves NP and PP in the target tree fragments. The rule below (also in Figure 1) is also a sentential form and matches since its (underlined) root labels of the target tree fragments read "NP PP". Thus, we can substitute the latter sentential form into the former and obtain the sentential form shown at the bottom of Figure 2. Ideally, the substitution process is repeated until the complete source sentence is derived.

4 Rule Extraction

The rule extraction of Maletti (2011) extracts minimal tree-to-tree rules, which are rules containing both source and target tree fragments, from sentence pairs of a word-aligned and bi-parsed parallel corpus. In particular, this requires parses for both the source and the target language sentences which adds a source for errors and specificity potentially leading to lower translation performance and lower coverage (Wellington et al., 2006). Chiang (2010) showed that string-to-tree systems—

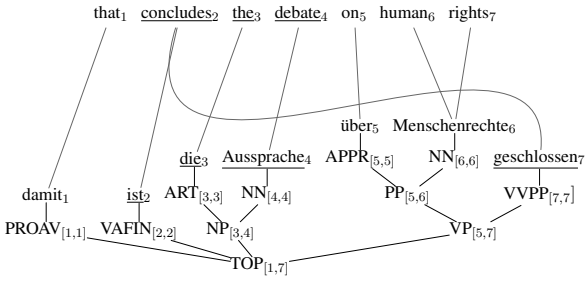

Figure 3: Word-aligned sentence pair with target-side parse.

which he calls fuzzy tree-to-tree-systems— generally yield higher translation quality compared to corresponding tree-to-tree systems.

For efficiency reasons the rule extraction of Maletti (2011) only extracts *minimal* rules, which are the smallest tree fragments compatible with the given word alignment and the parse trees. Similarly, non-minimal rules are those that can be obtained from minimal rules by substitution. In particular, each lexical item of a sentence pair occurs in exactly one minimal rule extracted from that sentence pair. However, minimal rules are especially unsuitable for fixed phrases consisting of rare words because minimal rules encourage small fragments and thus word-by-word translation. Consequently, such fixed phrases will often be assembled inconsistently by substitution from small fragments. Non-minimal rules encourage a consistent translation by covering larger parts of the source sentence.

Here we want to develop an efficient rule extraction procedure for our string-to-tree MBOTs that avoids the mentioned drawbacks. Naturally, we could substitute minimal rules into each other to obtain non-minimal rules, but performing substitution for all combinations is clearly intractable. Instead we essentially follow the approach of Koehn et al. (2003), Och and Ney (2004), and Chiang (2007), which is based on consistently aligned phrase pairs. Our training corpus contains *word-aligned sentence pairs* $\langle e, A, f\rangle$, which contain a source language sentence e, a target language sentence f, and an alignment $A \subseteq [1, \ell_e] \times [1, \ell_f]$, where ℓ_e and ℓ_f are the lengths of the sentences e and f, respectively, and $[i, i'] = \{j \in \mathbb{Z} \mid i \leq j \leq i'\}$ is the span (closed interval of integers) from i to i' for all positive integers $i \leq i'$. Rules are extracted for each pair of the corpus, so in the following let $\langle e, A, f\rangle$ be a word-aligned sentence pair. A *source phrase* is simply a span $[i, i'] \subseteq [1, \ell_e]$ and correspondingly, a *target phrase* is a span $[j, j'] \subseteq [1, \ell_f]$. A *rule span* is a pair $\langle p, \varphi\rangle$ consisting of a source phrase p and a sequence $\varphi = p_1 \cdots p_n$ of (non-overlapping) target phrases $p_1, \ldots, p_n$. Spans overlap if their intersection is non-empty. If $n = 1$ (i.e., there is exactly one target phrase in φ) then $\langle p, \varphi\rangle$ is also a phrase pair (Koehn et al., 2003). We want to emphasize that formally phrases are spans and not the substrings occuring at that span.

Next, we lift the notion of consistently aligned phrase pairs to our rule spans. Simply put, for a consistently aligned rule span $\langle p, p_1 \cdots p_n\rangle$ we require that it respects the alignment A in the sense that the origin i of an alignment $(i, j) \in A$ is covered by p if and only if the destination j is covered by $p_1, \ldots, p_n$. Formally, the rule span $\langle p, p_1 \cdots p_n\rangle$ is *consistently aligned* if for every $(i, j) \in A$ we have $i \in p$ if and only if $j \in \bigcup_{k=1}^{n} p_k$. For example, given the word-aligned sentence pair in Figure 3, the rule span $\langle [2, 4],\ [2, 4]\,[7, 7]\rangle$ is consistently aligned, whereas the phrase pair $\langle [2, 4],\ [2, 7]\rangle$ is not.

Our MBOTs use rules consisting of a source string and a sequence of target tree fragments. The target trees are provided by a parser for the target language. For each word-aligned sentence pair $\langle e, A, f\rangle$ we thus have a parse tree t for f. An example is provided in Figure 3. We omit a formal definition of trees, but recall that each node η of the parse tree t *governs* a (unique) target phrase. In Figure 3 we have indicated those target phrases (spans) as subscript to the non-lexical node labels. A consistently aligned rule span $\langle p, p_1 \cdots p_n\rangle$ of $\langle e, A, f\rangle$ is *compatible with* t if there exist nodes $\eta_1, \ldots, \eta_n$ of t such that η_k governs p_k for all $1 \leq k \leq n$. For example, given the word-aligned sentence pair and parse tree t in Figure 3, the consistently aligned rule span $\langle [2, 4],\ [2, 4]\,[7, 7]\rangle$ is not compatible with t because there is no node in t that governs $[2, 4]$. However, for the same data, the rule span $\langle [2, 4],\ [2, 2]\,[3, 4]\,[7, 7]\rangle$ is consistently aligned and compatible with t. The required nodes of t are labeled VAFIN, NP, VVPP.

Now we are ready to start the rule extraction. For each consistently aligned rule span $\langle p, p_1 \cdots p_n\rangle$ that is compatible with t and each selection of nodes $\eta_1, \ldots, \eta_n$ of t such that n_k governs p_k for each $1 \leq k \leq n$, we can extract the rule $e(p) \to \big(\mathrm{flat}(t_{\eta_1}), \ldots, \mathrm{flat}(t_{\eta_n})\big)$, where

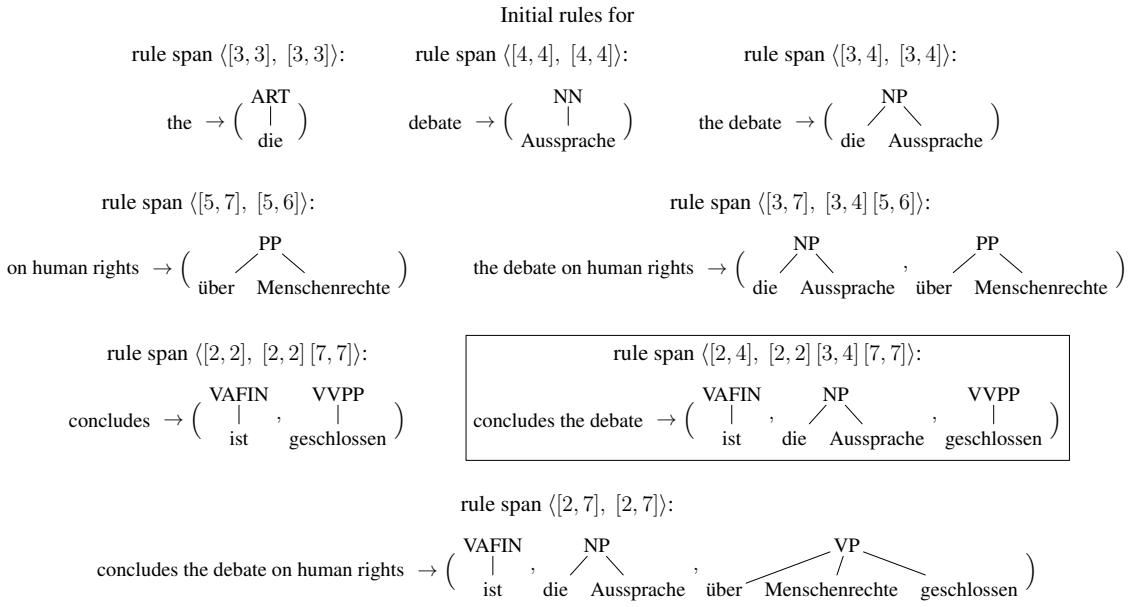

Figure 4: Some initial rules extracted from the word-aligned sentence pair and parse of Figure 3.

- $e(p)$ is the substring of e at span p,[1]
- $\mathrm{flat}(u)$ removes all internal nodes from u (all nodes except the root and the leaves), and
- t_η is the subtree rooted in η for node η of t.

The rules obtained in this manner are called *initial rules for* $\langle e, A, f\rangle$ *and* t. For example, for the rule span $\langle [2,4],\ [2,2]\,[3,4]\,[7,7]\rangle$ we can extract only one initial rule. More precisely, we have

- $e([2,4]) =$ concludes the debate
- $t_{\eta_1} =$ (VAFIN ist)
- $t_{\eta_2} =$ (NP (ART die) (NN Aussprache)),
- and $t_{\eta_3} =$ (VVPP geschlossen).

The function flat leaves t_{η_1} and t_{η_3} unchanged, but $\mathrm{flat}(t_{\eta_2}) =$ (NP die Aussprache). Thus, we obtain the boxed rule of Figure 4.

Clearly, the initial rules are just the start because they are completely lexical in the sense that they never contain the placeholder X in the source string nor a non-lexical leaf in any output tree fragment. We introduce non-lexical rules using the same approach as for the hierarchical rules of Chiang (2007). Roughly speaking, we obtain a new rule r'' by "excising" an initial rule r from another rule r' and replacing the removed part by

- the placeholder X in the source string,
- the root label of the removed tree fragment in the target tree fragments, and
- linking the removed parts appropriately,

so that the flatted substitution of r into r'' can yield r'. This "excision" process is illustrated in Figure 5, where we remove the middle initial rule from the topmost initial rule. The result is displayed at the bottom in Figure 5. Formally, the set of *extractable rules* R for a given word-aligned sentence pair $\langle e, A, f\rangle$ with parse tree t for f is the smallest set subject to the following two conditions:

- Each initial rule is in R and thus extractable.
- For every initial rule r and extractable rule $r' \in R$, any flat rule r'', into which we can substitute r to obtain ρ with $\mathrm{flat}(\rho) = r'$, is in R and thus extractable.[2]

For our running example depicted in Figure 3 we display some extractable rules in Figure 6.

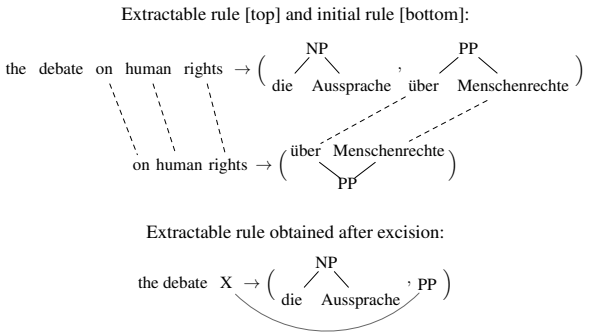

Figure 5: Excision of the middle initial rule from the topmost initial rule. Substituting the middle rule into the result yields the topmost rule.

[1] If $p = [i, i']$, then $e(p) = e[i, i']$ is the substring of e ranging from the i-th token to the i'-th token.

[2] A rule $\rho = s \to (t_1, \ldots, t_n)$ is *flat* if $\mathrm{flat}(\rho) = \rho$, where $\mathrm{flat}(\rho) = s \to (\mathrm{flat}(t_1), \ldots, \mathrm{flat}(t_n))$.

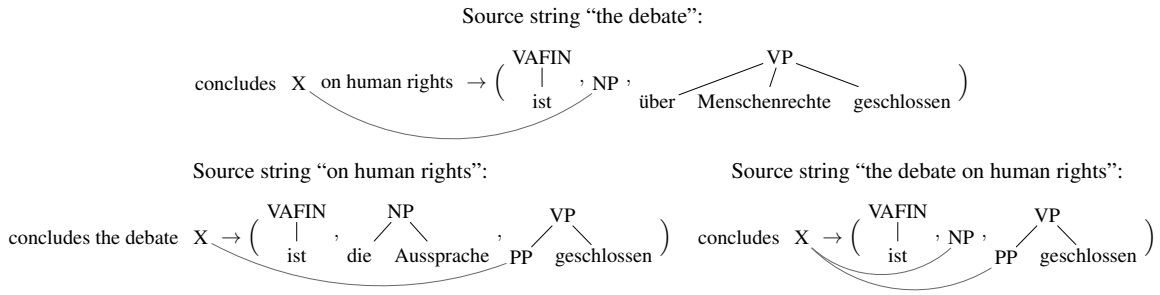

Figure 6: Extractable rules obtained by excising various initial rules (see Figure 4) from the initial rule displayed at the bottom of Figure 4.

Unfortunately, already Chiang (2007) points out that the set of all extractable rules is generally too large and keeping all extractable rules leads to slow training, slow decoding, and spurious ambiguity. Our MBOT rules are restricted by the parse tree for the target sentence, but the MBOT model permits additional flexibility due to the presence of multiple target tree fragments. Overall, we experience the same problems, and consequently, in the experiments we use the following additional constraints on rules $s \to (t_1, \ldots, t_n)$:

(a) We only consider source phrases p of length at most 10 (i.e., $i' - i < 10$ for $p = [i, i']$).[3]
(b) The source string s contains at most 5 occurrences of lexical items or X (i.e. $\ell_s \leq 5$).
(c) The source string s cannot have consecutive Xs (i.e., XX is not a substring of s).
(d) The source string contains at least one lexical item that was aligned in $\langle e, A, f \rangle$.
(e) The left-most token of the source string s cannot be X (i.e., $s[1, 1] \neq$ X).

Our implementation can easily be modified to handle other constraints. Figure 7 shows extractable rules violating those additional constraints.

Table 1 gives an overview on how many rules are extracted. Our string-to-tree variant extracts 12–17 times more rules than the minimal tree-to-tree rule extraction. For our experiments (see Section 6), we filter all rule tables on the given input. The decoding times for the minimal ℓMBOT and our MBOT share the same order of magnitude.

5 Model Features

For each source language sentence e, we want to determine its most likely translation $\hat{f}$ given by

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

[3]Note that this restricts the set of initial rules.

for some unknown probability distributions p. We estimate $p(e \mid f) \cdot p(f)$ by a log-linear combination of features $h_i(\cdot)$ with weights λ_i scored on sentential forms $e \to (t)$ of our extracted MBOT M such that the leaves of t read (left-to-right) f.

We use the decoder provided by MBOT-Moses of Braune et al. (2013) and its standard features, which includes all the common features (Koehn, 2009) and a gap penalty 100^{1-c}, where c is the number of target tree fragments that contributed to t. This feature discourages rules with many target tree fragments. As usual, all features are obtained as the product of the corresponding rule features for the rules used to derive $e \to (t)$ by means of substitution. The rule weights for the translation weights are obtained as relative frequencies normalized over all rules with the same right- and left-hand side. Good-Turing smoothing (Good, 1953) is applied to all rules that were extracted at most 10 times. The lexical translation weights are obtained as usual.

6 Experimental Results

We considered three reasonable baselines: (i) minimal ℓMBOT, (ii) non-contiguous STSSG (Sun et al., 2009), or (iii) a string-to-tree Moses system. We decided against the minimal ℓMBOT as a baseline since tree-to-tree systems generally get lower BLEU scores than string-to-tree systems. We nevertheless present its BLEU scores (see Table 3). Unfortunately, we could not compare to Sun et al. (2009) because their decoder and rule extraction algorithms are not publicly available. Furthermore, we have the impression that their system does not scale well:

- Only around 240,000 training sentences were used. Our training data contains between 1.8M and 5.7M sentence pairs.
- The development and test set were length-

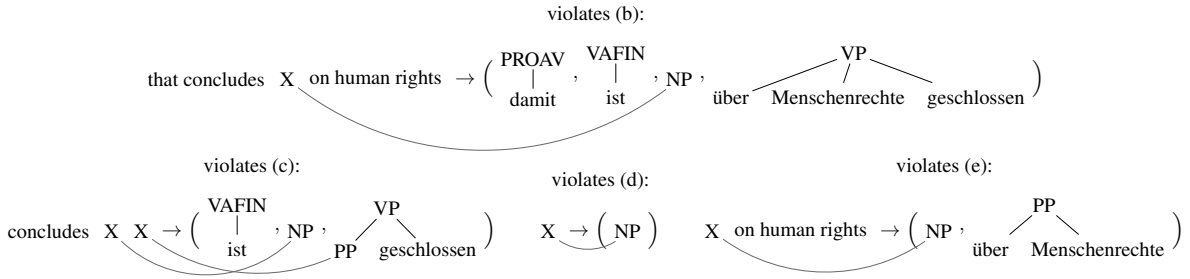

Figure 7: Showing extractable rules violating the restrictions.

System	number of extracted rules		
	English-To-German	English-To-Arabic	English-To-Chinese
minimal tree-to-tree ℓMBOT	12,478,160	28,725,229	10,162,325
non-minimal string-to-tree MBOT	143,661,376	491,307,787	162,240,663
string-to-tree Moses	14,092,729	55,169,043	17,047,570

Table 1: Overview of numbers of extracted rules with respect to the different extraction algorithms.

ratio filtered to sentences up to 50 characters. We do not modify those sets.

- Only rules with at most one gap were allowed which would be equivalent to restrict the number of target tree fragments to 2 in our system.

Hence we decided to use a string-to-tree Moses system as baseline (see Section 6.1).

6.1 Setup

As a baseline system for our experiments we use the syntax-based component (Hoang et al., 2009) of the Moses toolkit (Koehn et al., 2007). Our system is the presented translation system based on MBOTs. We use the MBOT-Moses decoder (Braune et al., 2013) which – similar to the baseline decoder – uses a CYK+ chart parsing algorithm using a standard X-style parse tree which is sped up by cube pruning (Chiang, 2007) with integrated language model scoring.

Our and the baseline system use linguistic syntactic annotation (parses) only on the target side (*string-to-tree*). During rule extraction we impose the restrictions of Section 4. Additional glue-rules that concatenate partial translations without performing any reordering are used in all systems.

For all experiments (English-to-German, English-to-Arabic, and English-to-Chinese), the training data was length-ratio filtered. The word alignments were generated by GIZA++ (Och and Ney, 2003) with the *grow-diag-final-and* heuristic (Koehn et al., 2005). The following language-specific processing was performed. The German text was true-cased and the functional and morphological annotations were removed from the parse. The Arabic text was tokenized with MADA (Habash et al., 2009) and transliterated according to Buckwalter (2002). Finally, the Chinese text was word-segmented using the Stanford Word Segmenter (Chang et al., 2008).

In all experiments the feature weights λ_i of the log-linear model were trained using minimum error rate training (Och, 2003). The remaining information for the experiments is presented in Table 2.

6.2 Quantitative Analysis

The overall translation quality was measured with 4-gram BLEU (Papineni et al., 2002) on true-cased data for German, on transliterated data for Arabic, and on word-segmented data for Chinese. Significance was computed with Gimpel's implementation (Gimpel, 2011) of pairwise bootstrap resampling with 1,000 samples. Table 3 lists the evaluation results. In all three setups the MBOT system significantly outperforms the baseline. For German we obtain a BLEU score of 15.90 which is a gain of 0.68 points. For Arabic we get an increase of 0.78 points which results in 49.10 BLEU. For Chinese we obtain a score of 18.35 BLEU gaining 0.66 points.[4] We also trained a vanilla phrase-based system for each language pair on the same data as described in Table 2.

To demonstrate the usefulness of the multiple

[4]NIST-08 also shows BLEU for word-segmented output (`http://www.itl.nist.gov/iad/mig/tests/mt/2008/doc/mt08_official_results_v0.html`). Best constrained system: 17.69 BLEU; best unconstrained system: 19.63 BLEU.

	English to German	English to Arabic	English to Chinese
training data	7th EuroParl corpus (Koehn, 2005)	MultiUN corpus (Eisele and Chen, 2010)	
training data size	$\approx$ 1.8M sentence pairs	$\approx$ 5.7M sentence pairs	$\approx$ 1.9M sentence pairs
target-side parser	BitPar (Schmid, 2004)	Berkeley parser (Petrov et al., 2006)	
language model	5-gram SRILM (Stolcke, 2002)		
add. LM data	WMT 2013	Arabic in MultiUN	Chinese in MultiUN
LM data size	$\approx$ 57M sentences	$\approx$ 9.7M sentences	$\approx$ 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 2: Summary of the performed experiments.

Language pair	System	BLEU
English-to-German	Moses Baseline	15.22
	MBOT	*15.90
	minimal ℓMBOT	14.09
	Phrase-based Moses	16.73
English-to-Arabic	Moses Baseline	48.32
	MBOT	*49.10
	minimal ℓMBOT	32.88
	Phrase-based Moses	50.27
English-to-Chinese	Moses Baseline	17.69
	MBOT	*18.35
	minimal ℓMBOT	12.01
	Phrase-based Moses	18.09

Table 3: Evaluation results. The starred results are statistically significant improvements over the baseline (at confidence $p < 1\%$).

target tree fragments of MBOTs, we analyzed the MBOT rules that were used when decoding the test set. We distinguish several types of rules. A rule is *contiguous* if it has only 1 target tree fragment. All other rules are (potentially) *discontiguous*. Moreover, *lexical* rules are rules whose leaves are exclusively lexical items. All other rules (i.e., those that contain at least one non-lexical leaf) are *structural*. Table 4 reports how many rules of each type are used during decoding for both our MBOT system and the minimal ℓMBOT. Below, we focus on analyzing our MBOT system. Out of the rules used for German, 27% were (potentially) discontiguous and 5% were structural. For Arabic, we observe 67% discontiguous rules and 26% structural rules. For translating into Chinese 30% discontiguous rules were used and the structural rules account to 18%. These numbers show that the usage of discontiguous rules tunes to the specific language pair. For instance, Arabic utilizes them more compared to German and Chinese. Furthermore, German uses a lot of lexical rules which is probably due to the fact that it is a morphologically rich language. On the other hand, Arabic and Chinese make good use of structural rules. In addition, Table 4 presents a finer-grained analysis based on the number of target tree fragments. Only rules with at most 8 target tree fragments were used. While German and Arabic seem to require some rules with 6 target tree fragments, Chinese probably does not. We conclude that the number of target tree fragments can be restricted to a language-pair specific number during rule extraction.

6.3 Qualitative Analysis

In this section, we inspect some English-to-German translations generated by the Moses baseline and our MBOT system in order to provide some evidence for linguistic constructions that our system handles better. We identified (a) the realization of reflexive pronouns, relative pronouns, and particle verbs, (b) the realization of verbal material, and (c) local and long distance reordering to be better throughout than in the baseline system. All examples are (parts of) translations of sentences from the test data. Ungrammatical constructions are enclosed in brackets and marked with a star. We focus on instances that seem relevant to the new ability to use non-minimal rules.

We start with an example showing the realization of a reflexive pronoun.

Source: Bitcoin *differs* from other types of virtual currency.
Reference: Bitcoin *unterscheidet sich* von anderen Arten virtueller Währungen.
Baseline: Bitcoin [*unterscheidet*]* von anderen Arten [der virtuellen Währung]*.

Language pair	System	Type	Lex	Struct	Total	Target tree fragments 2	3	4	5	≥ 6
English-to-German	our	cont.	27,351	635	27,986					
	MBOT	discont.	9,336	1,110	10,446	5,565	3,441	1,076	312	52
	minimal	cont.	55,910	4,492	60,402					
	ℓMBOT	discont.	2,167	7,386	9,553	6,458	2,589	471	34	1
English-to-Arabic	our	cont.	1,839	651	2,490					
	MBOT	discont.	3,670	1,324	4,994	3,008	1,269	528	153	36
	minimal	cont.	18,389	2,855	21,244					
	ℓMBOT	discont.	1,138	1,920	3,058	2,525	455	67	8	3
English-to-Chinese	our	cont.	17,135	1,585	18,720					
	MBOT	discont.	4,822	3,341	8,163	6,411	1,448	247	55	2
	minimal	cont.	34,275	8,820	43,095					
	ℓMBOT	discont.	516	4,292	4,808	3,816	900	82	6	4

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

MBOT: Bitcoin *unterscheidet sich* von anderen Arten [der virtuellen Währung]*.

Here the baseline drops the reflexive pronoun *sich*, which is correctly realized by the MBOT system. The rule used is displayed in Figure 8.

Figure 8: Rule realizing the reflexive pronoun.

Next, we show a translation in which our system correctly generates a whole verbal segment.

Source: *It turned out that* not only ...
Reference: *Es stellte sich heraus, dass* nicht nur ...
Baseline: [*Heraus,*]* nicht nur ...
MBOT: *Es stellte sich heraus, dass* nicht nur ...

The baseline drops the verbal construction whereas the large non-minimal rule of Figure 9 allows our MBOT to avoid that drop. Again, the required reflexive pronoun *sich* is realized as well as the necessary comma before the conjunction *dass*.

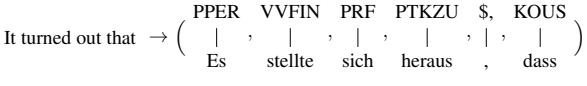

Figure 9: MBOT rule for the verbal segment.

Another feature of MBOT is its power to perform long distance reordering with the help of several discontiguous output fragments.

Source: ...weapons factories now, *which do not endure competition on the international market* and ...
Reference: ... Rüstungsfabriken, *die der internationalen Konkurrenz nicht standhalten* und ...
Baseline: ... [Waffen in den Fabriken nun]*, *die nicht einem Wettbewerb auf dem internationalen Markt* []* und ...
MBOT: ... [Waffen Fabriken nun]*, *die Konkurrenz auf dem internationalen Markt nicht ertragen* und ...

Figure 10 shows the rules which enable the MBOT system to produce the correct reordering.

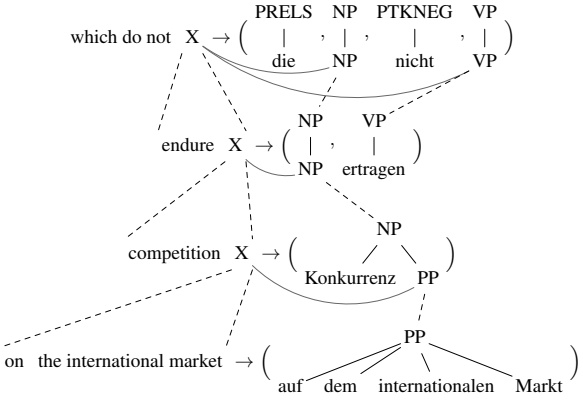

Figure 10: Long distance reordering.

7 Conclusion

We present an application of a string-to-tree variant of local multi bottom-up tree transducers, which are tree-to-tree models, to statistical machine translation. Originally, only minimal rules were extracted, but to overcome the typically lower translation quality of tree-to-tree systems and minimal rules, we abolish the syntactic annotation on the source side and develop a string-to-tree variant. In addition, we present a new pa-

rameterized rule extraction that can extract non-minimal rules, which are particularly helpful for translating fixed phrases. It would be interesting to know how much can be gained when using only one contribution at a time. Hence, we will explore the impact of string-to-tree and non-minimal rules in isolation.

We demonstrate that our new system significantly outperforms the standard Moses string-to-tree system on three different large-scale translation tasks (English-to-German, English-to-Arabic, and English-to-Chinese) with a gain between 0.53 and 0.87 BLEU points. An analysis of the rules used to decode the test sets suggests that the usage of discontiguous rules is tuned to each language pair. Furthermore, it shows that only discontiguous rules with at most 8 target tree fragments are used. Thus, further research could investigate a hard limit on the number of target tree fragments during rule extraction. We also perform a manual inspection of the obtained translations and confirm that our string-to-tree MBOT rules can adequately handle discontiguous phrases, which occur frequently in German, Arabic, and Chinese. Other languages that exhibit such phenomena include Czech, Dutch, Russian, and Polish. Thus, we hope that our approach can also be applied successfully to other language pairs.

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

Acknowledgement

The authors would like to express their gratitude to the reviewers for their helpful comments and Robin Kurtz for preparing the Arabic corpus.

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

References

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proc. 8th WMT*, pages 1–44. Association for Computational Linguistics.

Fabienne Braune, Nina Seemann, Daniel Quernheim, and Andreas Maletti. 2013. Shallow local multi bottom-up tree transducers in statistical machine translation. In *Proc. 51st ACL*, pages 811–821. Association for Computational Linguistics.

Timothy Buckwalter. 2002. Arabic transliteration. `http://www.qamus.org/transliteration.htm`.

Pi-Chuan Chang, Michel Galley, and Christopher D. Manning. 2008. Optimizing Chinese word segmentation for machine translation performance. In *Proc. 3rd WMT*, pages 224–232. Association for Computational Linguistics.

David Chiang. 2006. An introduction to synchronous grammars. In *Proc. 44th ACL*. Association for Computational Linguistics. Part of a tutorial given with Kevin Knight.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

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

Steve DeNeefe, Kevin Knight, Wei Wang, and Daniel Marcu. 2007. What can syntax-based MT learn from phrase-based MT? In *Proc. 2007 EMNLP*, pages 755–763. Association for Computational Linguistics.

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

Jason Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proc. 41st ACL*, pages 205–208. Association for Computational Linguistics.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proc. 2004 NAACL*, pages 273–280. Association for Computational Linguistics.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proc. 44th ACL*, pages 961–968. Association for Computational Linguistics.

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

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

Nizar Habash, Owen Rambow, and Ryan Roth. 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.

Hieu Hoang, Philipp Koehn, and Adam Lopez. 2009. A unified framework for phrase-based, hierarchical, and syntax-based statistical machine translation. In *Proc. 6th IWSLT*, pages 152–159.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. 2003 NAACL*, pages 48–54. Association for Computational Linguistics.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT Speech Translation Evaluation. In *Proc. 2nd IWSLT*, pages 68–75.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. 45th ACL*, pages 177–180. Association for Computational Linguistics.

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

Philipp Koehn. 2009. *Statistical Machine Translation*. Cambridge University Press.

Alon Lavie, Alok Parlikar, and Vamshi Ambati. 2008. Syntax-driven learning of sub-sentential translation equivalents and translation rules from parsed parallel corpora. In *Proc. 2nd SSST*, pages 87–95. Association for Computational Linguistics.

Yang Liu, Yajuan Lü, and Qun Liu. 2009. Improving tree-to-tree translation with packed forests. In *Proc. 47th ACL*, pages 558–566. Association for Computational Linguistics.

Andreas Maletti. 2011. How to train your multi bottom-up tree transducer. In *Proc. 49th ACL*, pages 825–834. Association for Computational Linguistics.

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

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

Franz J. Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. 40th ACL*, pages 311–318. Association for Computational Linguistics.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proc. 44th ACL*, pages 433–440. Association for Computational Linguistics.

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

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

Jun Sun, Min Zhang, and Chew Lim Tan. 2009. A non-contiguous tree sequence alignment-based model for statistical machine translation. In *Proc. 47th ACL*, pages 914–922. Association for Computational Linguistics.

Benjamin Wellington, Sonjia Waxmonsky, and I. Dan Melamed. 2006. Empirical lower bounds on the complexity of translational equivalence. In *Proc. 44th ACL*, pages 977–984. Association for Computational Linguistics.