Discontinuous Statistical Machine Translation with Target-Side Dependency Syntax

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

Abstract

For several languages only potentially non-projective dependency parses are readily available. Projectivizing the parses and utilizing them in syntax-based translation systems often yields particularly bad translation results indicating that those translation models cannot properly utilize such information. We demonstrate that our system based on multi bottom-up tree transducers, which can natively handle discontinuities, can avoid the large translation quality deterioration, achieve the best performance of all classical syntax-based translation systems, and close the gap to phrase-based and hierarchical systems that do not utilize syntax.

1 Introduction

Syntax-based machine translation, in which the transfer is achieved from and/or to the level of syntax, has become widely used in the statistical machine translation community (Bojar et al., 2014). Different grammar formalisms have been proposed and evaluated as translation models driving the translation systems. We use a variant of the local multi bottom-up tree transducer as proposed by Maletti (2011). More precisely, we use a *string-to-tree* variant of it, which offers two immediate advantages: (i) The source side of the rules is a simple string containing terminal symbols and the unique non-terminal X. Consequently, we do not need to match an input sentence parse, which allows additional flexibility. It has been demonstrated that this flexibility in the input often yields improved translation quality (Chiang, 2010). (ii) The target language side offers discontinuities because rules can contain a sequence of target tree fragments instead of a single tree fragment. These fragments are applied synchronously, which allows the model to synchronously develop discontinuous parts in the output (e.g., to realize agreement). Overall, this translation model already proved to be useful when translating from English into German, Chinese, and Arabic as demonstrated by Seemann et al. (2015). The goal of the current contribution is to adjust the approach and the system to Eastern European languages, for which we expect discontinuities to occur. The existing system (Seemann et al., 2015) cannot readily be applied since it requires constituent-like parses for the target side in our string-to-tree setting. However, for the target languages discussed here (Polish and Russian), only dependency parses are readily available. Those parses relate the lexical items of the sentence via edges that are labeled with the syntactic function between the head and its dependent. Overall, these structures also form trees, but they are often non-projective for our target languages. Such non-projective dependency trees do not admit a constituent-like tree representation, so we first need to convert them into projective dependency trees, which can be converted easily into a constituent-like tree representation. The conversion into projective dependency trees is known to preserve discontinuities, so we expect that out model is an ideally suited syntax-based translation model for those target languages.

We evaluate our approach in 2 standard translation tasks translating from English to both Polish and Russian. Those two target languages have rather free word order, so we expect discontinuities to occur frequently. For both languages, we use a (non-projective) dependency parser to obtain the required target trees, which we projectivize. Indeed, we confirm that non-projective parses are a frequent phenomenon in both languages. We then train our translation model on the constituent-like parse trees obtained from the projective dependency trees and evaluate the obtained machine translation systems. In both cases,

our system significantly outperforms the string-to-tree syntax-based component (Hoang et al., 2009) of MOSES. To put our evaluation scores into perspective, we also report scores for a vanilla phrase-based system (Och and Ney, 2004), a GHKM-based system (Galley et al., 2004), and a hierarchical phrase-based system (Chiang, 2007). It shows that our system suffers much less from the syntactic discontinuities and is thus much better suited for syntax-based translation systems in such settings.

2 Related work

Modern statistical machine translation systems (Koehn, 2009) are build using various different translation models as their core. Syntax-based systems are widely used nowadays due to their innate ability to handle non-local reordering and other linguistic phenomena. For certain language pairs they even outperform phrase-based models (Och and Ney, 2004) and constitute the state-of-the-art (Bojar et al., 2014). Our MBOT is a variant of the shallow local multi bottom-up tree transducer presented by Braune et al. (2013). Alternative models include the synchronous tree substitution grammars of Eisner (2003), which use a single source and target tree fragment per rule. Our MBOT rules similarly contain a single source tree fragment, but a sequence of target tree fragments. The latter feature enables discontinuous translations. Another model that offers this feature for the source and the target language side is the non-contiguous synchronous tree-sequence substitution grammar of Sun et al. (2009), which offers sequences of tree fragments on both sides.

The idea of utilizing dependency trees in machine translation is not novel. Bojar and Hajič (2008) built a system based on synchronous tree substitution grammars for English-to-Czech that uses projective dependency trees. Xie et al. (2011) present a dependency-to-string model that extracts head-dependent rules with reordering information. Their model requires a custom decoder to deal with the dependency information in the input. Li et al. (2014) follow up on this work by transforming these dependency trees into (a kind of) constituency trees. In this approach, they are able to use the conventional syntax-based models of MOSES. In contrast to our work, these two models do not use the syntactic functions provided by the parser but rather extract head-dependent rules based on the lexical items. Sennrich et al. (2015) transformed (non-projective) dependency trees into constituency trees using the syntactic functions provided by the parser. They used the string-to-tree GHKM model (Williams and Koehn, 2012) of MOSES and evaluated their approach on an English-to-German translation task. It shows that the system utilizing the (transformed) dependency parses outperforms competing systems utilizing various variants of constituent parses for the German side. We follow up on their work for translation tasks, where constituent parses are not readily available, and achieve translation quality that is comparable to phrase-based systems for two language pairs (English-to-Polish and English-to-Russian).

3 Transformation of Dependency Trees into Constituency Trees

In this section, we present a short overview of dependency parsing and introduce the non-projective tree structures that occur as parses. We need to transform these structures into projective trees, which are then converted into the shape of classical constituency trees.

3.1 Description

The syntax of languages with relatively free word order, which includes Polish and Russian, is often difficult to express in terms of constituency structure (Kallestinova, 2007). Since the parts that need to (grammatically) agree can occur spread out over the whole sentence, constituents cannot be hierarchically organized as in a classical constituency parse tree. Dependency parses do not pre-suppose such a hierarchical structure and are thus often more suitable for languages with free word order.

In a dependency parse each occurrence of a lexical item (i.e., token) in the input sentence forms a node. The dependency parser constructs a tree structure over those nodes by relating them via edges pointing from a *head* node h to its *dependent* node d. Such an edge is denoted by $h \rightarrow d$. In addition, each edge is assigned a label indicating the type of the syntactic dependence. Often an artificial root node is added for convenience. An example parse for a Polish sentence is depicted in Figure 1.

Next, we distinguish between projective and

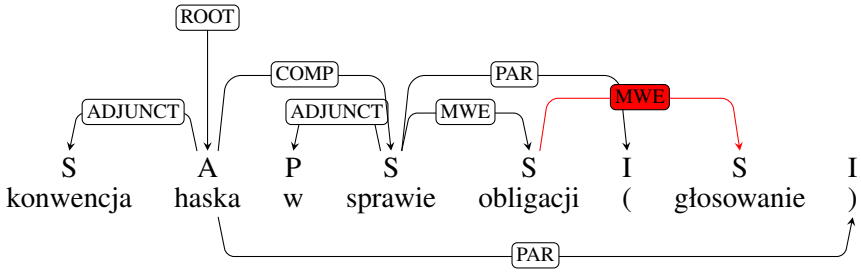

Figure 1: Non-projective Polish dependency tree [gloss: *hague convention on securities (vote)*].

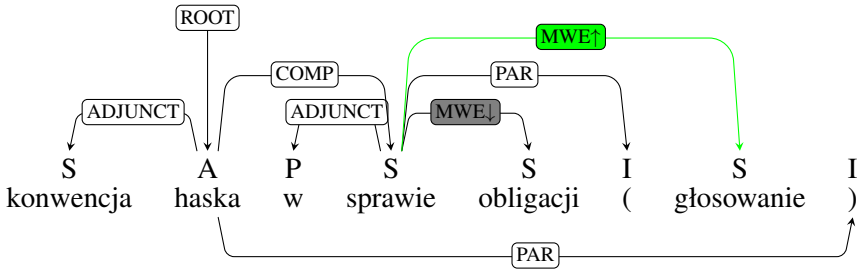

Figure 2: Projective dependency parse obtained by 'path'-lifting.

non-projective edges. The edge $h \rightarrow d$ is *projective* if and only if its head node h dominates[1] all nodes representing the tokens in the linear span between h and d. For example, the edge 'obligacji $\rightarrow$ głosowanie' is non-projective because 'obligacji' does not dominate '(', which occurs in the relevant linear span. A dependency parse is projective if and only if all its edges are projective. A non-projective dependency parse is easily recognized in graphical representations because it has a crossing edge provided that all the edges are drawn on one side of the sentence as in Figure 1.

Non-projective dependency structures cannot be directly used in the translation framework MOSES (Koehn et al., 2007), so we first have to turn them into projective trees. To this end, Kahane et al. (1998) came up with the idea of *lifting*. Given a non-projective edge $h \rightarrow d$ there exists (at least) one node n that occurs in the linear span between h and d such that n is not dominated by h. In the lifting process, the edge $h \rightarrow d$ is replaced by an edge $g \rightarrow d$, where g is the lowest node that dominates both h and n (i.e., the least common ancestor of h and n). Repeating this process for all non-projective edges eventually yields a projective tree. Nivre and Nilsson (2005) refined this approach and introduced three additional ways of lifting: 'head', 'head+path', and 'path', which perform the same replacement but annotate different information in the labels to document the lifting process. The annotation schemes 'head' and 'head+path' might increase the number of labels quadratically, whereas 'path' only introduces a linear number of new labels. Since we deal with millions of trees in our syntax-based machine translation experiments, we need to select a compromise between (i) inflating the number of labels and (ii) documenting the lifts. We decided to use the 'path' scheme to obtain projective parse trees for our experiments (see Section 5).

Let us explain the 'path' scheme. In the situation described earlier, in which the edge $h \rightarrow d$ was replaced by the edge $g \rightarrow d$, we set the label of $g \rightarrow d$ to the label of the original edge $h \rightarrow d$ annotated by $\uparrow$ to indicate that this edge was lifted. Additionally, all edges connecting the new head g and the syntactical head h are annotated with $\downarrow$ indicating where the syntactic head is found. Figure 2 shows the projective tree obtained from the non-projective parse of Figure 1. In it we have the new edge 'sprawie $\rightarrow$ głosowanie' with label 'MWE$\uparrow$'. Moreover, the edge 'sprawie $\rightarrow$ obligacji' now has the label MWE$\downarrow$ because it is the edge that connects the new head with the syntactical head of 'głosowanie'.

In principle, one can imagine other ways to projectivize a tree; e.g., we can just replace the head

[1] A node n *dominates* a node d iff n is an ancestor of d; i.e., there is a path from n to d.

of a non-projective edge by the root. From a linguistic point of view, it makes more sense to attach it (as described) to the least common ancestor, which in a sense is the minimal required change that leaves the remaining edges in place. Furthermore, the used implementation always lifts the most nested[2] non-projective edge until the tree is projective. In this way, the minimal number of lifts required to projectivize the tree is achieved as demonstrated by Buch-Kromann (2005).

3.2 Implementation

We aim to investigate string-to-tree machine translation systems, so we need syntactic annotations on the target side. First, the target-side sentences (in Polish and Russian) are annotated with part-of-speech tags with the help of TREETAGGER (Schmid, 1994). The TREETAGGER output is then converted into the (comma-separated) CONNL-X format[3], which lists each token of the sentence in one line with 10 attributes like word position, word form, lemma, and part-of-speech tag. A new sentence is started by an empty line. This representation is passed to the MALT parser (Nivre et al., 2006; Sharoff and Nivre, 2011), which fills the remaining attribute fields like position of the head and the label of dependency edges. The resulting output represents the (potentially) non-projective dependency parses of the target-side sentences.

In the next step, we apply the 'path'-lifting as described in Section 3.1. In total, we performed 500,507 lifts for Polish (corpus size: 14,147,378 tokens) and 137,893 lifts for Russian (corpus size: 30,808,946 tokens) to make the corresponding parses projective. As described in Section 3.1 we introduce at most 3 additional labels for each existing label. In Table 1 we report for each corpus the exact number of original parse labels and the number of labels newly introduced by the transformation into projective parses.

Finally, we transform the projective dependency parse trees directly into the standard representation of constituent parse trees in MOSES.[4] We use the part-of-speech tags as pre-terminal nodes. Additionally, we make the labels and part-of-speech tags more uniform as follows:

[2]deepest or most distant from the root

[3]documented on http://ilk.uvt.nl/conll/

[4]http://www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/mbotmoses.html

Corpus	Lang.	Number of labels original	new
EUROPARL	PL	25	67
YANDEX	RU	75	118
Commoncrawl	RU	71	84
News commentary	RU	71	84
Patronymic names	RU	13	0
Names	RU	31	0
WIKI headlines	RU	54	19

Table 1: Number of parse labels before and after the 'path'-lifting.

- All parentheses are labeled 'PAR'.
- All slashes, quotation marks, and dashes are labeled 'PUNCT' and their part-of-speech tag is 'INTJ'.
- All punctuation marks are labeled 'PUNC' and their part-of-speech tag is ','.
- If the tagger did not assign a part-of-speech tag, then we label it 'UNK'.

The final constituency tree representation obtained from the projective dependency tree of Figure 2 is shown in Figure 3.

4 Translation Model

We use the string-to-tree variant (Seemann et al., 2015) of the multi bottom-up tree transducer (Maletti, 2010) as translation model. For simplicity, we call the variant 'MBOT'. A more detailed discussion of the model can be found in (Seemann et al., 2015; Maletti, 2011). Let us attempt a high-level description. An MBOT is a synchronous grammar (Chiang, 2006) that is similar to a synchronous context-free grammar. Instead of a single source and target fragment in each rule, MBOT rules are of the form $s \to (t_1, \ldots, t_n)$ containing a single *source string* s and potentially several *target tree fragments* $t_1, \ldots, t_n$. The source string is built from the lexical items and the special placeholder X, which can also occur several times. Each occurrence of X is linked to some non-lexical leaves in the target tree fragments. In contrast to most synchronous grammars, each placeholder occurrence can link to several leaves in the target tree fragments indicating that these parts are supposed to develop synchronously. However, each non-lexical leaf in the target tree fragments links to exactly one placeholder occurrence (see top rule in Figure 4). A finite set of such rules constitutes an MBOT. Several rules of an MBOT for trans-

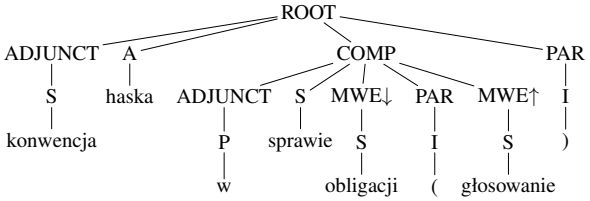

Figure 3: Final constituency representation for the parse of Figure 2.

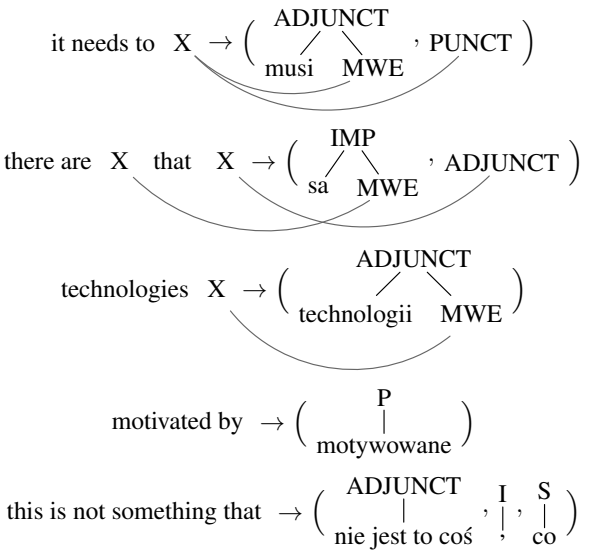

Figure 4: Several rules of an MBOT.

lating from English (source) to Polish (target) are shown in Figure 4. The bottom rule is both lexical and discontinuous. Note that it can be used in a continuous manner, but it is as well possible to plug additional material between the three target tree fragments.

The rules were extracted with the method described and the implementation provided by Seemann et al. (2015). The standard log-linear model (Koehn, 2009) is used with the following features:

(1) forward translation weight
(2) indirect translation weight
(3) forward lexical translation weight
(4) indirect lexical translation weight
(5) target-side language model
(6) word penalty
(7) rule penalty
(8) gap penalty 100^{1-c}, where c is the number of target tree fragments used in the derivation of the output tree.

All those features are standard except for the gap penalty, which is intended to discourage derivations that involve large numbers of target tree fragments, thus providing a feature to favor or disfavor continuous derivations. As usual, the (forward and indirect) translation weights are obtained as products of corresponding rule weights, which are obtained by maximum likelihood estimation. All rules that were extracted at most 10 times are smoothed using GOOD-TURING smoothing (Good, 1953). Both lexical translation weights are obtained from the co-occurrence statistics obtained during word alignment. The standard decoder of MBOT-MOSES by Braune et al. (2013) is used to generate translations using our model. As in the standard syntax-based component (Hoang et al., 2009), this decoder is a CYK+ chart parser based on standard X-style parse trees with integrated language model scoring that is accelerated by cube pruning (Chiang, 2007).

5 Experimental Results

We evaluate the MBOT-based system (see Section 4) on two translation tasks: English-to-Polish and English-to-Russian. For both target languages only (potentially) non-projective dependency parses are easily available. Our goal is to evaluate whether the discontinuity offered by the MBOT model helps in tasks involving such dependency parses. Consequently, the baseline system is the syntax-based component (Hoang et al., 2009) of the MOSES toolkit (Koehn et al., 2007), which uses a translation model that only permits continuous rules. Both systems are *string-to-tree* in the sense that the projectivized parses are only used on the target side. As mentioned in Section 3, the non-projective parses are obtained using the MALT parser and then converted to constituent-like trees. Glue-rules in both systems ensure that partial translation candidates can always be concatenated without any reordering.

5.1 Setup

We use standard and freely available resources to build our machine translation systems. In summary, for Russian we use the resources provided by the 2014 Workshop on Statistical Machine Translation (Bojar et al., 2014). The Polish data is taken from the EUROPARL corpus (Koehn, 2005).

Next, let us describe the preparation and evaluation for both tasks (English-to-Polish and English-to-Russian). An overview of the used resources is presented in Table 2. First, the training data was

	English to Polish	English to Russian
training data size	≈ 618K sentence pairs	≈ 1.7M sentence pairs
target-side parser	Malt parser (Nivre et al., 2006; Sharoff and Nivre, 2011)	
parser grammar	(Wróblewska and Przepiórkowski, 2012)	(Nivre et al., 2008)
language model (LM)	5-gram SRILM (Stolcke, 2002)	
additional LM data	Polish sentences in EuroParl	WMT 2014
LM data size	≈ 626K sentences	≈ 43M sentences
development test size	3,030 sentences	3,000 sentences
test size	3,029 sentences	3,003 sentences

Table 2: Summary of the experimental setup.

length-ratio filtered, tokenized, and lowercased. We used GIZA++ (Och, 2003) with the 'grow-diag-final-and' heuristic (Koehn et al., 2005) to automatically derive the word alignments. The feature weights of the log-linear models were trained with the help of minimum error rate training (Och and Ney, 2003) and optimized for 4-gram BLEU (Papineni et al., 2002) on the development test set (lowercased, tokenized). In the end, the systems were evaluated (also using 4-gram BLEU) on the test set. Significance judgments of the differences in the reported translation quality (as measured by BLEU) were computed with the pairwise bootstrap resampling technique of Koehn (2004) on 1,000 samples. Table 2 summarizes the setup information.

A particular detail is worth mentioning. The authors were unable to identify standard development and test sets for the English-to-Polish translation task. Consequently, we manually removed one session of the EUROPARL corpus. After removing duplicate sentences, we used the odd numbered sentences as development set and the even numbered sentences as test set.

5.2 Analysis

We present the quantitative evaluation for both experiments in Table 3. In both cases (English-to-Polish and English-to-Russian) the MBOT system significantly outperforms the baseline, which is the syntax-based component of MOSES. For Polish we obtain a BLEU score of 23.43 resulting in a gain of 2.14 points over the baseline. Similarly, for Russian we achieve a BLEU score of 26.13, which is an increase of 1.47 points over the baseline. To put our results in perspective, we also trained a GHKM system, a phrase-based system, and a hierarchical phrase-based system (Hiero) with standard settings for each translation task on the same resources as described in Table 2 and present their evaluation also in Table 3.

Translation task	System	BLEU
English-to-Polish	Baseline	21.29
	MBOT	**23.43**
	GHKM	23.31
	Phrase-based	24.35
	Hiero	24.56
English-to-Russian	Baseline	24.66
	MBOT	**26.13**
	GHKM	25.97
	Phrase-based	27.90
	Hiero	27.72

Table 3: Evaluation results incl. MOSES phrase-based system, GHKM-based system, and hierarchical system for reference. The bold MBOT results are statistically significant improvements over the baseline (at confidence $p < 1\%$).

Based on the observed BLEU scores, it seems likely that our MBOT-based approach can almost completely avoid the large quality drop observed between a (hierarchical) phrase-based system, which does not utilize the syntactic annotation, and a continuous string-to-tree syntax-based model. The availability of discontinuous tree fragments yields significant improvements in translation quality (as measured by BLEU) and an overall performance similar to (hierarchical) phrase-based systems. However, we also observe that outscoring a (hierarchical) phrase-based remains a challenge, so it remains to be seen whether syntactic information can actually help the translation quality in those translation tasks.

To quantitatively support our claim that the multiple target tree fragments (and the discontinuity) of an MBOT are useful, we provide statistics on the MBOT rules that were used to decode the test set. To this end, we distinguish several types of rules. A rule is *continuous* if it has only 1 target tree fragment, and all other rules are (potentially) *discontinuous*. Additionally, we distinguish *lexical* rules, which only contain lexical items as leaves, and *structural* rules, which contain at least one non-lexical leaf. In Table 4 we report how many rules of each type are used during decoding.[5]

For Polish, 41% of all used rules were discontinuous and only 4% were structural. Similarly, 35% of the used Russian rules were discontinuous and again only 4% were structural. The low proportion of structural rules is not very surprising since both languages are known to be morphologically rich and thus have large lexicons (167,657 lexical items in Polish and 911,397 lexical items in Russian). Another interesting point is the distribution of *discontinuous structural rules*. Polish and Russian use 83% and 62%, respectively, showing that the majority of the used structural rules is discontinuous in both tasks. Additionally using the data of Seemann et al. (2015), we can confirm that morphologically rich languages have a small minority of structural rules (4%, 4%, and 5% for Polish, Russian, and German, respectively), whereas Arabic and Chinese use a much larger proportion of structural rules (26% and 18%, respectively). In addition, we suspect that the additional non-projectivity of Polish makes discontinuous rules more useful (as an indicator for induced discontinuity). Whereas for Russian, German, Arabic and Chinese approx. 2 out of 3 used structural rules are discontinuous (62%, 64%, 67%, and 68%, respectively), more than 4 out of 5 (83%) used structural rules are discontinuous for Polish.

Finally, we present a fine-grained analysis based on the number of target tree fragments in Table 4. Useful Polish rules have at most 6 target tree fragments, whereas Russian rules with up to 9 target tree fragments have been used. Similar numbers have been reported in (Seemann et al., 2015). Using their data, we also note that Polish, Russian, and Chinese seem to use a larger percentage of discontinuous rules with 2 output tree fragments (80%–90%) compared to German and Arabic (50%–60%).

[5]The provided analysis tools currently do not support an analysis whether a discontinuous rule was actually used in a discontinuous manner or whether the components were later combined in a continuous manner. The reported numbers thus represent potential discontinuity.

6 Conclusion

We presented an application of string-to-tree local multi bottom-up tree transducers as translation model of a syntax-based machine translation system. The obtained system uses rules with a string on the source language side and a sequence of target tree fragments on the target language side. The availability of several target tree fragments in a single rule enables the model to realize discontinuous translations. We expected that particularly translation into languages with discontinuous constituents would benefit from our model. However, such languages often have rather free word order and often only dependency parsers are available for them. The mentioned discontinuities often produce non-projective parses, which we need to transform into projective constituent-like parse trees before they can be utilized in MOSES. Hence, we (i) applied a lifting technique to projectivize the dependency trees, which stores information about the performed lift operations in the new labels, and (ii) transformed the obtained projective dependency trees into constituent-like trees.

Next, we demonstrated that the discontinuous string-to-tree system significantly outperforms the standard MOSES string-to-tree system on two different translation tasks (English-to-Polish and English-to-Russian) with large gains of 2.14 and 1.47 BLEU points, respectively. We also trained a vanilla phrase-based system, a GHKM-based system, and a hierarchical system for each translation task. In comparison to the string-to-string phrase-based system, the discontinuous string-to-tree system is only 0.92 BLEU points worse on English-to-Polish and 1.77 BLEU points worse for English-to-Russian. It thus remains to be seen whether machine translation systems can benefit from syntactic information in those translation tasks, but the proposed model at least avoids the large quality drop observed for the continuous string-to-tree system.

Finally, we analyzed the rules used by our system to decode the test sets. In summary, it shows that both our target languages (Polish and Russian) require a lot of lexical rules, which is most

Translation task	Type	Lex	Struct	Total	Target tree fragments 2	3	4	5	≥ 6
English-to-Polish	cont.	25,327	307	25,634					
	discont.	16,312	1,595	17,907	15,805	1,818	254	27	3
English-to-Russian	cont.	24,100	664	24764					
	discont.	12,767	1,108	13,875	11,087	2,308	412	58	10

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

likely due to the morphological richness of the languages. Furthermore, they use a lot of discontinuous structural rules, which confirms our assumption that a system allowing discontinuous target tree fragments is the right choice for such languages.

Acknowledgment

The authors would like to express their gratitude to the reviewers for their helpful comments. Furthermore, we would like to thank ANDERS BJÖRKELUND and WOLFGANG SEEKER for their shared expertise on dependency parsing.

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

References

Ondřej Bojar and Jan Hajič. 2008. Phrase-based and deep syntactic English-to-Czech statistical machine translation. In *Proc. 3rd WMT*, pages 143–146. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proc. 9th WMT*, pages 12–58. 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.

Matthias Buch-Kromann. 2005. *Discontinuous Grammar — A dependency-based model of human parsing and language learning*. Ph.D. thesis, Copenhagen Business School.

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.

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. NAACL*, pages 273–280. Association for Computational Linguistics.

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

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. ISCA.

Sylvain Kahane, Alexis Nasr, and Owen Rambow. 1998. Pseudo-projectivity: A polynomially parsable non-projective dependency grammar. In *Proc. 36th ACL*, pages 646–652. Association for Computational Linguistics.

Elena Dmitrievna Kallestinova. 2007. *Aspects of Word Order in Russian*. Ph.D. thesis, University of Iowa, IA, USA.

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. ISCA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran,

Richard Zens, Chris Dyer, Ondřej 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. 2004. Statistical significance tests for machine translation evaluation. In *Proc. EMNLP*, pages 388–395. 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.

Liangyou Li, Jun Xie, Andy Way, and Qun Liu. 2014. Transformation and decomposition for efficiently implementing and improving dependency-to-string model in Moses. In *Proc. 8th SSST*, pages 122–131. Association for Computational Linguistics.

Andreas Maletti. 2010. Why synchronous tree substitution grammars? In *Proc. HLT-NAACL*, pages 876–884. 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.

Joakim Nivre and Jens Nilsson. 2005. Pseudo-projective dependency parsing. In *Proc. 43rd ACL*, pages 99–106. Association for Computational Linguistics.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *Proc. 5th LREC*, pages 2216–2219. European Language Resources Association.

Joakim Nivre, Igor M. Boguslavsky, and Leonid L. Iomdin. 2008. Parsing the SYNTAGRUS treebank of Russian. In *Proc. 22nd CoLing*, pages 641–648. Association for Computational Linguistics.

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.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proc. Int. Conf. New Methods in Language Processing*, pages 44–49. University of Manchester, Institute of Science and Technology.

Nina Seemann, Fabienne Braune, and Andreas Maletti. 2015. String-to-tree multi bottom-up tree transducers. In *Proc. 53rd ACL*, pages 815–824. Association for Computational Linguistics.

Rico Sennrich, Philip Williams, and Matthias Huck. 2015. A tree does not make a well-formed sentence: Improving syntactic string-to-tree statistical machine translation with more linguistic knowledge. *Computer Speech & Language*, 32(1):27–45.

Serge Sharoff and Joakim Nivre. 2011. The proper place of men and machines in language technology processing Russian without any linguistic knowledge. In *Proc. Dialogue*, pages 657–670. Russian State University for the Humanities.

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

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.

Philip Williams and Philipp Koehn. 2012. GHKM rule extraction and scope-3 parsing in Moses. In *Proc. 7th WMT*, pages 388–394. Association for Computational Linguistics.

Alina Wróblewska and Adam Przepiórkowski. 2012. Induction of dependency structures based on weighted projection. In *Proc. 4th ICCCI*, volume 7653 of LNAI, pages 364–374. Springer.

Jun Xie, Haitao Mi, and Qun Liu. 2011. A novel dependency-to-string model for statistical machine translation. In *Proc. EMNLP*, pages 216–226. Association for Computational Linguistics.