Hierarchical Machine Translation With Discontinuous Phrases

Miriam Kaeshammer
University of Düsseldorf
Universitätsstraße 1
40225 Düsseldorf, Germany
kaeshammer@phil.uni-duesseldorf.de

Abstract

We present a hierarchical statistical machine translation system which supports discontinuous constituents. It is based on synchronous linear context-free rewriting systems (SLCFRS), an extension to synchronous context-free grammars in which synchronized non-terminals span $k \geq 1$ continuous blocks on either side of the bitext. This extension beyond context-freeness is motivated by certain complex alignment configurations that are beyond the alignment capacity of current translation models and their relatively frequent occurrence in hand-aligned data. Our experiments for translating from German to English demonstrate the feasibility of training and decoding with more expressive translation models such as SLCFRS and show a modest improvement over a context-free baseline.

1 Introduction

In statistical machine translation, phrase-based translation models with a beam search decoder (Koehn et al., 2003) and tree-based models with a CYK decoder represent two prominent types of approaches. The latter usually employ some form of synchronous context-free grammar (SCFG). They can be grouped into so-called hierarchical phrase-based models that are formally syntax-based, such as in Chiang (2007), and models where hierarchical units are somehow linguistically motivated, e.g. in Zollmann and Venugopal (2006) and Hoang and Koehn (2010).

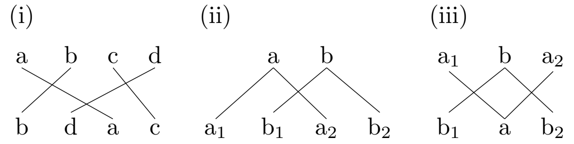


Figure 1: Complex alignment configurations: (i) inside-out alignment; (ii) CDTU; (iii) bonbon. The configurations can also occur upside down.

The adequacy of all of these models has been questioned, as the space of alignments that they generate is limited. Inside-out alignments are beyond the alignment capacity of SCFG of rank 2 (henceforth 2-SCFG) and inversion transduction grammar (Wu, 1997), but they can be generated with phrase-based translation models thanks to the reordering component of standard decoders. Cross-serial discontinuous translation units (CDTU) (Søgaard and Kuhn, 2009) and bonbon configurations (Simard et al., 2005) in contrast can neither be generated by a phrase-based translation system nor by an SCFG-based one. It is thereby assumed that a translation unit, the transitive closure of a set of nodes of the bipartite alignment graph, represents minimal translational equivalence, and therefore that an adequate translation grammar formalism should be able to generate each translation unit separately.

The aforementioned problematic alignment configurations are schematically depicted in Figure 1. Alignment (i) is an inside-out alignment; it is formed by four translation units (a, b, c and d). CDTUs (ii) and bonbons (iii) each consist of two intertwined discontinuous translation units.

Several studies have investigated the alignment capacity of SCFG-based and phrase-based translation models in different setups (Wellington et al., 2006; Søgaard and Kuhn, 2009; Søgaard and Wu, 2009; Søgaard, 2010; Kaeshammer, 2013). For example, Wellington et al. (2006) find that inside-out alignments occur in 5% of their manually aligned English-Chinese sentence pairs. In the study of Kaeshammer (2013), 9% of the sentence pairs in a Spanish-French data set and 5.5% of the sentence pairs in an English-German data set cannot be generated by a 2-SCFG. In addition, Kaes-

Proceedings of the Tenth Workshop on Statistical Machine Translation, pages 228–238,
Lisboa, Portugal, 17-18 September 2015. ©2015 Association for Computational Linguistics.

hammer and Westburg (2014) qualitatively investigate the instances of the complex alignment configurations in the same English-German data set and find that even though some of them are due to annotation errors, most of them are correctly annotated phenomena that one would like to be able to generate when translating.

To be able to induce the alignment configurations in question, more expressive translation models and corresponding decoding algorithms are necessary. For the phrase-based models, Galley and Manning (2010) propose a translation model that uses discontinuous phrases and a corresponding beam search decoder. For tree-based models, a grammar formalism beyond the power of context-free grammar is necessary. Søgaard (2008) proposes to apply range concatenation grammar; Kaeshammer (2013) puts forward the idea of using synchronous linear context-free rewriting systems (SLCFRS), a direct extension of SCFG to discontinous constituents. To the best of our knowledge, neither of the two proposals have resulted in an actual machine translation system.

With this work, we extend the line of research proposed in Kaeshammer (2013), and present the first full tree-based statistical machine translation system that allows for discontinuous constituents. It is thus able to produce the complex alignment configurations in Figure 1. As such, it combines the advantage of being able to learn and generate discontinuous phrases with the benefits of tree-based translation models.

Currently, our system is hierarchical phrase-based, i.e. it does not make use of linguistically motivated syntactic annotation. However, it will be straightforward to transfer methods to integrate linguistic constituency information from the SCFG-based machine translation literature (such as Zollmann and Venugopal (2006)) to our approach. This is particularly interesting, since, in the monolingual parsing community, approaches that are able to produce constituency trees with discontinuous constituents have become increasingly popular (Maier, 2010; van Cranenburgh and Bod, 2013; Kallmeyer and Maier, 2013). Recently, such parsers have reached a speed with which it would actually be feasible to parse the training set of a machine translation system (Versley, 2014; Maier, 2015; Fernández-González and Martins, 2015), which is necessary to train syntactically motivated translation grammars.

In this work, we define a translation model based on SLCFRS, explain the training of a corresponding hierarchical phrase-based grammar, provide details about a corresponding decoder and results of experiments for translating from German to English.

2 Model

Our translation model is a weighted synchronous LCFRS. Conceptually, this grammar formalism is very close to synchronous CFG, with the addition that non-terminals span tuples of strings (instead of just strings) on either side of the bitext. Just as SCFGs, an SLCFRS can be used for synchronous parsing of parallel sentences as well as for translating monolingual sentences. For the latter, the source side of the synchronous grammar is used to parse the input text, thereby generating target side derivations from which the translations can be read off.

2.1 Synchronous LCFRS

An LCFRS[1] (Vijay-Shanker et al., 1987; Weir, 1988) is a tuple $G = (N, T, V, P, S)$ where N is a finite set of non-terminals with a function dim: $N \rightarrow \mathbb{N}$ determining the *fan-out* of each $A \in N$; T and V are disjoint finite sets of terminals and variables; $S \in N$ is the start symbol with $dim(S) = 1$; and P is a finite set of rewriting rules

$$A(\alpha_1, \ldots, \alpha_{dim(A)}) \rightarrow A_1(Y_1^{(1)}, \ldots, Y_{dim(A_1)}^{(1)}) \\ \cdots A_m(Y_1^{(m)}, \ldots, Y_{dim(A_m)}^{(m)})$$

where $A, A_1, \ldots, A_m \in N$, $Y_j^{(i)} \in V$ for $1 \leq i \leq m$, $1 \leq j \leq dim(A_i)$ and $\alpha_i \in (T \cup V)^*$ for $1 \leq i \leq dim(A)$, for a *rank* $m \geq 0$. For all $r \in P$, it holds that every variable Y in r occurs exactly once in the left-hand side (LHS) and exactly once in the right-hand side (RHS) of r.

A non-terminal is instantiated with respect to some input string w such that terminals and variables are consistently mapped to w. A rule r explains how an instantiated LHS non-terminal can be rewritten by its instantiated RHS non-terminals. A derivation starts with the start symbol S instantiated to the input string w. All strings that can

[1] We use the syntax of simple range concatenation grammars (Boullier, 1998), an equivalent formalism.

$$\begin{array}{ll}
\langle A(a,c) \rightarrow \varepsilon & , \; C(a,c) \rightarrow \varepsilon \rangle \\
\langle B(b,d) \rightarrow \varepsilon & , \; D(bd) \rightarrow \varepsilon \rangle \\
\langle A(aX,cZ) \rightarrow A_{\boxed{1}}(X,Z) & , \; C(aX,Zc) \rightarrow C_{\boxed{1}}(X,Z) \rangle \\
\langle B(bY,dU) \rightarrow B_{\boxed{1}}(Y,U) & , \; D(bYd) \rightarrow D_{\boxed{1}}(Y) \rangle \\
\langle S(XYZU) \rightarrow A_{\boxed{1}}(X,Z)B_{\boxed{2}}(Y,U) \, , & \\
& S(XYZ) \rightarrow C_{\boxed{1}}(X,Z)D_{\boxed{2}}(Y) \rangle
\end{array}$$

Figure 2: Rules of an SLCFRS for $L = \{\langle a^n b^m c^n d^m, a^n b^m d^m c^n \rangle \mid n, m > 0\}$, taken from Kaeshammer (2013).

$$\begin{array}{l}
\langle S_{\boxed{1}}(aabccd), S_{\boxed{1}}(aabdcc) \rangle \\
\quad \Rightarrow \langle A_{\boxed{2}}(aa,cc)B_{\boxed{3}}(b,d), C_{\boxed{2}}(aa,cc)D_{\boxed{3}}(bd) \rangle \\
\quad \Rightarrow \langle A_{\boxed{2}}(aa,cc), C_{\boxed{2}}(aa,cc) \rangle \\
\quad \Rightarrow \langle A_{\boxed{4}}(a,c), C_{\boxed{4}}(a,c) \rangle \\
\quad \Rightarrow \varepsilon
\end{array}$$

Figure 3: Derivation of $\langle aabccd, aabdcc \rangle$ using the rules in Figure 2.

be rewritten to ε are in the language of the grammar. For more formal definitions, see for example Kallmeyer (2010).

The *rank* of a grammar G is the maximal rank of any of its rules, and its *fan-out* is the maximal fan-out of any of its non-terminals. G is called a (u, v)-LCFRS if it has rank u and fan-out v. A CFG is the special case of an LCFRS with fan-out $v = 1$. An LCFRS is *monotone* if, for every rule and every RHS non-terminal, the order of the variables in the arguments of this non-terminal is the same as the order of these variables in the arguments of the LHS non-terminal of this rule. This means that the order of (instantiated) arguments of the LHS non-terminal of a rule always corresponds to their order in the input sentence. An LCFRS is called *ε-free* if all of its rules in P are ε-free, which means that none of their LHS arguments is the empty string ε.[2]

The definition of synchronous LCFRS (SLCFRS) follows the definition of synchronous CFG, as for example in Satta and Peserico (2005). An SLCFRS (Kaeshammer, 2013) is a tuple $G = (N_s, N_t, T_s, T_t, V_s, V_t, P, S_s, S_t)$ where N_s, T_s, V_s, S_s, resp. N_t, T_t, V_t, S_t are defined as for LCFRS. They denote the alphabets for the *source* and *target side* respectively. P is a finite set of synchronous rewriting rules $\langle r_s, r_t, \sim \rangle$ where r_s and r_t are LCFRS rewriting rules based on N_s, T_s, V_s and N_t, T_t, V_t respectively, and $\sim$ is a bijective mapping of the non-terminals in the RHS of r_s to the non-terminals in the RHS of r_t. This link relation is represented by co-indexation in the synchronous rules. During a derivation, the yields of two co-indexed non-terminals have to be explained from one synchronous rule. $\langle S_s, S_t \rangle$ is the start pair. In such a derivation, we call the yield of S_s the *source side yield* and the yield of S_t the *target side yield*. SLCFRS are equivalent to simple range concatenation transducers (Bertsch and Nederhof, 2001).

Figure 2 shows an example. The synchronous rules translate cross-serial dependencies into nested ones. A sample derivation is shown in Figure 3.

The tuple $(N_s, T_s, V_s, P_s, S_s)$ is called the *source side grammar G_s* and $(N_t, T_t, V_t, P_t, S_t)$ the *target side grammar G_t*, where P_s is the set of all r_s in P and P_t is the set of all r_t in P. The *rank u* of a SLCFRS G is the maximal rank of G_s and G_t, and the *fan-out v* of G is the sum of the fan-outs of G_s and G_t. One may write $v_{v_{G_s}|v_{G_t}}$ to make clear how the fan-out of G is distributed over the source and the target side. As in the monolingual case, a corresponding grammar G is called a (u, v)-SLCFRS. The rank of the corresponding grammar in Figure 2 is 2 and its fan-out $4_{2|2}$. We call an SLCFRS *monotone* if the source side grammar as well as the target side grammar is monotone. We call an SLCFRS *ε-free* if the source side grammar as well as the target side grammar is ε-free.

We further define some terms which will be used in the following sections. A *range* in a string w_1^n is a pair $\langle l, r \rangle$ with $0 \leq l \leq r \leq n$. Its *yield* $\langle l, r \rangle(w)$ is the string w_{l+1}^r. The yield of a vector of ranges $\boldsymbol{\rho}(w)$ is the vector of the yields of the single ranges.

2.2 Definition

Given a source sentence f and an SLCFRS, generally, many derivations will have f as the source side yield, leading to many (different) target side yields, i.e. possible translations e. As it is standard in statistical machine translation, we use a log-linear model over derivations D to weight those translation options. The definition closely follows the model definition for SCFG, see Chiang (2007)

[2] An LCFRS is also ε-free if it contains a rule $S(\varepsilon) \rightarrow \varepsilon$, but S does not appear in any RHS of the rules in P.

for example.

$$\begin{aligned} P(D) &\propto \prod_i \phi_i(D)^{\lambda_i} \\ &\propto P_{LM}(e)^{\lambda_{LM}} \cdot w(D) \end{aligned}$$

where ϕ_i are features defined on the derivations, and λ_i are feature weights to be set during tuning. An n-gram language model provides a feature $P_{LM}(e)$ for the probability of seeing the target sentence e as derived by D. The other features ($i \neq LM$) are defined on the rules of a weighted SLCFRS which are used in the derivation D.

A weighted SLCFRS is an SLCFRS that is additionally equipped with a weight function w which assigns a weight to each synchronous rule $r \in P$. To fit the log-linear model, we define w as

$$w(r) = \prod_{i \neq LM} \phi_i(r)^{\lambda_i}$$

The weight of a derivation D is then

$$w(D) = \prod_{r \in D} w(r)$$

2.3 Features

We use the following standard features $\phi_i(r)$:

- translation probabilities in both directions $P(r_s|r_t)$ and $P(r_t|r_s)$,
- lexical weights $lex(r_s|r_t)$ and $lex(r_t|r_s)$ (Koehn et al., 2003) that estimate how well the terminals in the rule translate to each other,
- a rule penalty $\exp(1)$,
- a word penalty $\exp(-|w_t|)$ where $|w_t|$ is the number of terminals that occur in r_t.

In addition, we devise features that characterize the amount of expressivity beyond context-freeness of the applied rules. The *source gap degree* of r is the fan-out of r_s minus 1, and the *target gap degree* of r is the fan-out of r_t minus 1. See Maier and Lichte (2011) for more details about gap degree. These features can be read off the rules r directly. They allow the model to learn a preference for or against using the more powerful rules.

We also use glue rules, as proposed by Chiang (2005), which allow for a monotone combination of synchronous constituents as in a phrase-based model. A glue rule feature of value $\exp(1)$ with its weight λ_{glue} controls their usage.

3 Training

The synchronous rules are extracted from a corpus of parallel sentences that have already been word-aligned. Following Och and Ney (2004) and Chiang (2005), we extract all rules that are consistent with the word alignment A of a sentence pair $\langle f, e \rangle$ in a two-step procedure. First, *initial phrase pairs* are extracted; they correspond to terminal rules. Second, hierarchical rules are created by replacing phrase pairs that are contained within other phrase pairs with non-terminals/variables.

The crucial difference to previous work on translation with SCFG is that initial phrases do not have to be continuous. Instead, a phrase is a set of word indices, as in Galley and Manning (2010). Given $\langle f, e \rangle$ and a corresponding word alignment A, a phrase pair $(\bar{s}, \bar{t})$ is consistent with A if the following holds:

$$\begin{aligned} &\forall (i,j) \in A : i \in \bar{s} \leftrightarrow j \in \bar{t} \\ &\wedge\ \exists i \in \bar{s}, j \in \bar{t} : (i,j) \in A \end{aligned}$$

For each initial phrase pair $(\bar{s}, \bar{t})$, a terminal synchronous rule of the following form is created and added to P:

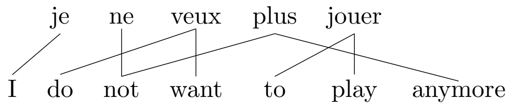
$$\langle X(\boldsymbol{\rho}_s(f)) \rightarrow \varepsilon, X(\boldsymbol{\rho}_t(e)) \rightarrow \varepsilon \rangle$$

$\boldsymbol{\rho}_s$ and $\boldsymbol{\rho}_t$ are range vectors, applied to the source sentence f and target sentence e respectively. $\boldsymbol{\rho}_s$ (respectively $\boldsymbol{\rho}_t$) is obtained by partitioning $\bar{s}$ (respectively $\bar{t}$) such that each subset contains all and only consecutive indices, designating a continuous block of the discontinuous phrase. Such a subset X is turned into a range $\langle l, r \rangle$ with $l = \min(X)$ and $r = \max(X)$. The ranges obtained from $\bar{s}$ (respectively $\bar{t}$), in ascending order, form $\boldsymbol{\rho}_s$ (respectively $\boldsymbol{\rho}_t$).

Furthermore, if P contains a rule $\langle X(\boldsymbol{\alpha}) \rightarrow \boldsymbol{\Psi}, X(\boldsymbol{\beta}) \rightarrow \boldsymbol{\Theta} \rangle$ that has been built from a phrase pair $(\bar{s}, \bar{t})$ and the set of phrase pairs contains a pair $(\bar{s}', \bar{t}')$ such that $\bar{s}' \subset \bar{s}$ and $\bar{t}' \subset \bar{t}$, we add the following new rule to P:

$$\begin{aligned} &\langle X(\boldsymbol{\alpha}') \rightarrow \boldsymbol{\Psi} X_{\boxed{k}}(Y_1, \ldots, Y_{h_s}), \\ &X(\boldsymbol{\beta}') \rightarrow \boldsymbol{\Theta} X_{\boxed{k}}(Z_1, \ldots, Z_{h_t}) \rangle \end{aligned}$$

A new non-terminal X is added to the RHS of r_s and r_t. k is an index that is not yet used in the bijective mapping of non-terminals in $\boldsymbol{\Psi}$ and $\boldsymbol{\Theta}$. Range vectors $\boldsymbol{\rho}_{s'}$ and $\boldsymbol{\rho}_{t'}$ are deduced from $\bar{s}'$ and $\bar{t}'$ as described above. Each range in $\boldsymbol{\rho}_{s'}$ (respectively $\boldsymbol{\rho}_{t'}$) is associated with a variable Y_i for



Initial phrase pairs:

1. jouer — to play
2. veux — do . . . want
3. ne veux plus — do not want . . . anymore
4. ne veux plus jouer — do not want to play anymore

. . .

Rules:

1. $\langle X(\text{jouer}) \to \varepsilon,\ X(\text{to play}) \to \varepsilon\rangle$
2. $\langle X(\text{veux}) \to \varepsilon,\ X(\text{do, want}) \to \varepsilon\rangle$
3. $\langle X(\text{ne veux plus}) \to \varepsilon,\ X(\text{do not want, anymore}) \to \varepsilon\rangle$
4. $\langle X(\text{ne veux plus jouer}) \to \varepsilon,$
   $X(\text{do not want to play anymore}) \to \varepsilon\rangle$
5. $\langle X(\text{ne } Y_1 \text{ plus}) \to X_{\boxed{1}}(Y_1),$
   $X(Z_1 \text{ not } Z_2, \text{anymore}) \to X_{\boxed{1}}(Z_1, Z_2)\rangle$
6. $\langle X(\text{ne veux plus } Y_1) \to X_{\boxed{1}}(Y_1),$
   $X(\text{do not want } Z_1 \text{ anymore}) \to X_{\boxed{1}}(Z_1)\rangle$
7. $\langle X(\text{ne } Y_1 \text{ plus } Y_2) \to X_{\boxed{1}}(Y_1)X_{\boxed{2}}(Y_2),$
   $X(Z_1 \text{ not } Z_2 Z_3 \text{ anymore}) \to X_{\boxed{1}}(Z_1, Z_2)X_{\boxed{2}}(Z_3)\rangle$

. . .

Figure 4: Sample rules that are extracted from the provided aligned sentence pair.

$1 \le i \le h_s$ (respectively Z_j for $1 \le j \le h_t$), where h_s (respectively h_t) is the length of $\boldsymbol{\rho}_{s'}$ (respectively $\boldsymbol{\rho}_{s'}$). They have to be variables that are not yet in use in $\boldsymbol{\alpha}$ (respectively $\boldsymbol{\beta}$). Those variables constitute the arguments of the new synchronous non-terminal X. Accordingly, h_s and h_t are the fan-outs of X on the source and the target side respectively. $\boldsymbol{\alpha}'$ (respectively $\boldsymbol{\beta}'$) is created from $\boldsymbol{\alpha}$ (respectively $\boldsymbol{\beta}$) by replacing the terminals that correspond to ranges in $\boldsymbol{\rho}_{s'}$ (respectively $\boldsymbol{\rho}_{t'}$) with the variable Y_i (respectively Z_j) that as been associated to the range. Note that this extraction yields only monotone and ε-free (S)LCFRS, which simplifies parsing.

The discontinuous rule extraction procedure is exemplified in Figure 4. Rule #5 for example was created from rule #3 by substituting phrase pair #2. Note that phrase pairs #1 and #4 are also extracted by a phrase-based system, and rules #1, #4 and #6 are also generated by a hierarchical phrase-based, i.e. SCFG-based, system. Rule #6 would usually be written down as

$$X \to \langle \text{ne veux plus } X_{\boxed{1}}, \text{do not want to } X_{\boxed{1}} \text{ anymore}\rangle$$

However, just as Galley and Manning (2010), we extract many more rules that also capture discontinuous translation units. In addition, we also extract rules which are discontinuous and hierarchical at the same time. They capture relationships between possibly discontinuous translation units.

Enumerating all discontinuous phrase pairs is exponential in the maximum phrase length. Therefore, in addition to the constraints that are generally set for SCFG extraction (e.g. phrase length, number of non-terminals, adjacent non-terminals on the source side, unaligned words at phrase edges, see Chiang (2007)), we also restrict the number of words that can be in a gap, we disallow unaligned blocks, and we restrict the number of continuous blocks in a phrase to 2. The latter is motivated by the results presented in Kaeshammer (2013) where a fan-out of $4_{2|2}$ is enough to derive the alignments in all data sets. We furthermore analyse the alignments of the training data before running the extraction and only allow discontinuous phrase pairs in synchronous spans which contain any of the alignment configurations that are beyond the power of SCFG.

As derivations are not observable in the training data, we use the method described in Chiang (2007) to hypothesize a distribution based on the counts of the extracted rules and then use relative-frequency estimation to obtain $P(r_s|r_t)$ and $P(r_t|r_s)$.

4 Decoder

Our decoder closely follows the methodology of current SCFG decoders, with the difference that it is able to handle source and target discontinuities in the form of SLCFRS rules. The goal is to find the target sequence e of the highest scoring derivation D according to the model defined in Section 2.2 that yields $\langle f, e\rangle$, where f is the given input sentence.

We parse the input sentence with a bottom-up CYK parser using the source side of the SLCFRS translation grammar. This corresponds to monolingual probabilistic LCFRS parsing, which has been described for example in Kallmeyer and Maier (2013). Using the rules, parse items are built. They are of the form $[A, \boldsymbol{\rho}]$, where A is a non-terminal label and $\boldsymbol{\rho}$ is a range vector indicating which part of the input is covered by this item. For the label, we use a combination of the source side label and the target side label in order to ensure valid target side derivations. Smaller items, i.e. items that cover less input words, are created before larger items. Equal items are combined, thereby retaining their origin via hyperedges.

When creating a new item using a specific rule, the variables and arguments in the rule have to be replaced consistently with ranges $\langle l, r \rangle$ of the input sentence. Roughly, this means that terminals and variables are instantiated with ranges such that for ranges that are adjacent in an argument of the LHS non-terminal, the concatenation of the two ranges has to be defined, i.e. $r_1 = l_2$ for $\langle l_1, r_1 \rangle$ and $\langle l_2, r_2 \rangle$. For example, given the input ${}_0\mathit{il}_1\mathit{ne}_2\mathit{mange}_3\mathit{plus}_4$, $X(\langle 1,4 \rangle) \rightarrow X(\langle 2,3 \rangle)$ is an instantiation of the source side of rule #5 from Figure 4. We can make further assumptions about rule instantiations, as our rules are all monotone, ε-free and we do not allow for empty gaps to avoid spurious ambiguity.

In the implementation, we first replace all terminals with all possible ranges with respect to the input sentence in an initialisation step; for instance $X(\langle 1,2 \rangle\, Y_1\, \langle 3,4 \rangle) \rightarrow X(Y_1)$ for the previous example. During the actual parsing, we are then only concerned with how variables are instantiated. We implement different pruning methods, such as limiting the number of target side rules for the same source side rule, and limiting the number of incoming hyperedges for one parse item.

Because of the specific form of the grammar that we have extracted (rank 2, fan-out $4_{2|2}$), we implement a specific parser for $(2,2)$-LCFRS. Accordingly, the range vector $\boldsymbol{\rho}$ of an item has the form $\langle \langle i_1, j_1 \rangle, \langle i_2, j_2 \rangle \rangle$, where i_2 and j_2 are undefined if the yield of the item is continuous. Such range vectors can be stored and retrieved more efficiently than general range vectors, i.e. for full LCFRS (which are typically implemented as bit vectors of the size of the input sentence). Also parsing time complexity is directly dependent on the fan-out v_s of the monolingual grammar: $\mathcal{O}(|G_s| \cdot |f|^{v_s \cdot (u+1)})$ with rank $u = 2$ and fan-out $v_s = 2$ in our case.

Finally, the parse hypergraph that we obtain from parsing with the source side of the grammar is intersected with an n-gram language model to also integrate $P_{LM}(e)$. We use cube pruning for this step (Chiang, 2007; Huang and Chiang, 2007). The difference to SCFG-based implementations is that the target string of a hypothesis that is scored by the language model is not necessarily continuous, but consists of a tuple of continuous blocks of target words, e.g. $\langle$do not want, anymore$\rangle$ if we would like to score a hypothesis which has been built from rule #3 in Figure 4. Therefore, each continuous block is scored separately and contributes its score to the overall score of the hypothesis. Furthermore, we need to store one language model state (simply put remembering the first and last $n-1$ words of the block) for each block. This means that a language model state in our implementation is a vector of conventional language model states of the length of the size of the target tuple of the hypothesis. Note that since our grammar has a target fan-out of 2, this vector has a maximal length of 2, but this is not a fixed limit in the implementation.

Since obtaining the k-best translations for a given input sentence is essential for tuning, we implement k-best extraction on the hypergraph that we obtain after cube pruning. We adopt the lazy strategy from Huang and Chiang (2005).

The decoder is implemented in C++, including code from KenLM[3] for language modelling.

5 Experiments

5.1 Setup

We run experiments for German-to-English, based on data that has been used in the WMT 2014 translation task[4]. For training of the translation models, we use the parallel sentences from Europarl and the News Commentary Corpus up to a length of 30 words (1.3M sentence pairs). For language modeling, we use the KenLM Language Model Toolkit[5]. We train a 3-gram language model on all available monolingual English data (Europarl, News Commentary, News Crawl, 92.7M sentences). From the available development data, we use `newstest2013` as the development test set (max. 25 words). From the rest, we randomly select 3000 sentence pairs of a maximal length of 25 words as development set. We further refine this set to sentences without out-of-vocabulary source words by decoding the development set once and selecting the corresponding sentences. We thus end up with 1694 sentence pairs for tuning. As our test set, we use the cleaned test set that has been made available (2280 sentence pairs with a maximal length of 30 words).

[3] http://kheafield.com/code/kenlm/developers/

[4] http://www.statmt.org/wmt14/translation-task.html

[5] http://kheafield.com/code/kenlm/

We normalize the punctuation, tokenize and truecase all our data using the scripts that are available in Moses[6] (Koehn et al., 2007). Furthermore, we perform compound splitting for German, also with the script provided in Moses.

The training data is word-aligned by running multi-threaded GIZA++ in both directions and then symmetrizing the alignments using the `grow-diag-final-and` heuristics as implemented in the Moses training script (step 1–4). Lexical translation probabilities are also emitted as part of this pipeline. For grammar extraction, we limit the length of initial phrases and the number of words in a gap to 10. We neither allow unaligned words at edges of initial phrases nor unaligned blocks.

Before decoding a data set with our decoder, we filter the large translation grammar with respect to the input data by extracting per-sentence-grammars. These only contain rules whose terminals match the words in the sentence to translate.

For the reported results, we set the buffer size for cube pruning to 400. We do not limit the number of words a non-terminal can span. We neither restrict the number of incoming hyperedges for the parse items nor the number of target side rules for the same source side rule.

Tuning the feature weights is done with minimum error rate training (Och, 2003), maximizing BLEU-4 (Papineni et al., 2002) and using the 200 best translations. For our own decoder, we use the very flexible implementation Z-MERT v1.50 (Zaidan, 2009). For Moses, we use the provided tuning script `mert-moses.pl`.

All reported BLEU scores have been calculated with the Moses script `multi-bleu.perl`, using the lowercase option `-lc`. Because of the variance that is introduced by tuning, we repeated each experiment four times and report the average of the final BLEU scores as well as the standard deviation.

5.2 Results

We compare different versions of our system against each other. The baseline is a system which uses only SCFG rules, i.e. a hierarchical phrase-based system. We refer to it as SYS(1,1), as it uses an SLCFRS of fan-out $2_{1|1}$. SYS(1,2) is a system which uses a grammar of fan-out $3_{1|2}$, i.e. it builds only continuous constituents on the source side, but allows for discontinuous constituents with two blocks on the target side. SYS(2,1) is the analogous system which restricts the target side to continuous constituents. Finally, SYS(2,2) uses an SLCFRS of fan-out $4_{2|2}$.

		devtest		test	
system	feat	BLEU	std	BLEU	std
SYS(1,1)	-	24.13	0.10	23.23	0.11
SYS(1,2)	-	23.39	0.32	23.24	0.09
SYS(2,1)	-	24.17	0.09	**23.41**	0.06
SYS(2,2)	-	23.90	0.13	22.90	0.03
SYS(2,2)	S	24.06	0.23	23.17	0.19
SYS(2,2)	T	**24.20**	0.15	23.35	0.04
SYS(2,2)	S+T	24.18	0.20	23.32	0.13
MOSES		24.33	0.08	23.34	0.20

Table 1: Averaged BLEU scores over four tuning runs; the feat column indicates whether additional source/target gap degree features have been used

Table 1 displays the main results. Allowing gaps on the source and the target side (SYS(2,2)) leads to a decline in BLEU score compared to the baseline. We hypothesize that this is due to weak probability estimates because of data sparseness and the additional ambiguity that is caused by the new rules with discontinuities. However, when adding the features about the gap degree of the rules used in the derivation, the model has an additional way of influencing which kind of rules are used. Especially controlling for the target gap degree turns out to be important and leads to a small improvement in BLEU score. Note, however, that rules with target gaps are not totally dismissed when this feature is switched on. Usage of rules with a target gap goes down from on average 734.5 rules in SYS(2,2) to on average 76.5 rules in SYS(2,2)-T in the test set. They are used less often, but, it seems, in a more controlled and sensible way.

This tendency is further confirmed with the experiments in which the discontinuous rules are only used on one side. While restricting the source side derivations to continuous yields does not improve the BLEU score (it rather severely degrades it in the case of the devtest set), restricting the target side derivations leads to a small improvement in BLEU score, and even to the best system for the test set. This is in particular interesting with respect to translation times since restricting the target side to continuous yields means removing the additional complexity that target gaps mean for the

[6] `http://www.statmt.org/moses/`

	SYS(1,1)	SYS(2,1)	=
e1	43	49	3
e2	46	47	2

Table 2: Result of the manual system comparison

		e2		
		SYS(1,1)	SYS(2,1)	=
	SYS(1,1)	**29**	13	1
e1	SYS(2,1)	15	**33**	1
	=	2	1	0

Table 3: Confusion matrix of the decisions of the manual evaluation

language model integration (see Section 4).

We also report results for the hierarchical phrase-based system in Moses trained on the same data as our systems. We tried to use the same settings as for our comparable system SYS(1,1). However, given the number of parameters during training and decoding, the various interpretations thereof and numerous implementation details to consider, it is not too surprising that the Moses system actually produces different translations than ours. The reported numbers merely serve as a point of reference, indicating that the translations produced by our system are not totally far off.

5.3 Manual Evaluation

We furthermore performed a manual evaluation in form of a system comparison using our own installation of the Appraise tool (Federmann, 2012). We compare the baseline SYS(1,1) against SYS(2,1), the best-performing setup on the test set. For each of them, we randomly selected one of the four configurations that lead to the reported averaged BLEU score. We then selected those translations of the test set where SYS(2,1) uses at least one SLCFRS rule with a discontinuity (95 sentences).

We asked two native speakers of English (e1, e2) with basic knowledge of German to evaluate our test sentences. They were shown the source sentence, a reference translation, the SYS(1,1) translation and the SYS(2,1) translation. The latter two were presented anonymized and in random order. The options for the evaluators were (a) translation A is better than B, (b) translation B is better than A, and (c) translations A and B are of equal quality. We specifically asked them to use option (c) as rarely as possible.

Table 2 shows the results. While our human evaluators do not demonstrate a clear preference for one of the systems, there is, however, a slight preference for the system that uses discontinuous rules (SYS(2,1)). In spite of the inter-annotator agreement being not very high (Cohen's $\kappa = 0.338$), the tendency for SYS(2,1) is also perceivable for the translations for which the evaluators agree in their decisions, see Table 3.

5.4 Translation Example

We finish this section with an actual translation example. It is picked because it makes crucial use of the discontinuous SLCFRS rules. It is taken from the test set.

In Figure 5, the following rule, which has a fan-out of 2 on the source side, leads to an overall grammatical sentence structure and a meaningful translation:

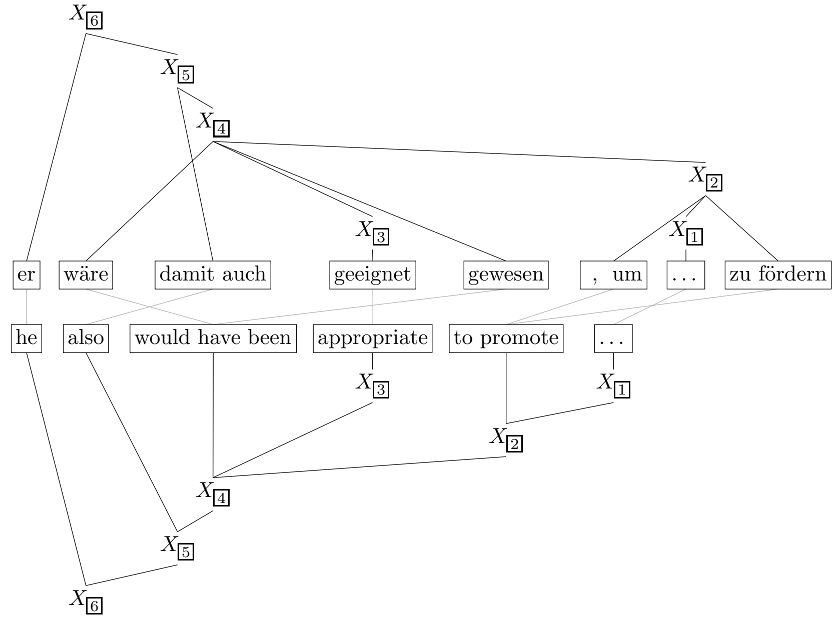
$$\langle\, X(\text{wäre}\ , Y_1\ \text{gewesen}\ Y_2) \rightarrow X_{\boxed{1}}(Y_1)X_{\boxed{2}}(Y_2)\,,$$
$$X(\text{would have been}\ Y_1 Y_2) \rightarrow X_{\boxed{1}}(Y_1)X_{\boxed{2}}(Y_2)\,\rangle$$

The rule derives the synchronous constituent labelled $X_{\boxed{4}}$ in Figure 5. Besides providing a correct verbal translation in a specific tense, it also establishes a relationship to the adjective ($X_{\boxed{1}}$) and the infinitive subordinate clause ($X_{\boxed{2}}$), thereby still leaving room for the adverb in terms of the gap on the source side. The adverb is then introduced with the following rule, leading to the constituent labelled $X_{\boxed{5}}$ in Figure 5:

$$\langle\, X(Y_1\ \text{damit auch}\ Y_2) \rightarrow X_{\boxed{1}}(Y_1, Y_2)\,,$$
$$X(\text{also}\ Y_1) \rightarrow X_{\boxed{1}}(Y_1)\,\rangle$$

This rule can be seen as capturing the different placement of the adverb *auch*/*also* in German and English.

Note that the alignment that is induced by the SYS(2,1) derivation is also derivable with a $2_{1|1}$-SLCFRS. One general possibility is to allow rules of rank $u > 2$. Another possibility is to put the individual phrases together in a different order and hierarchy. For example, in an SCFG rule, the discontinuous verb phrase could be combined with the adjective and the adverb first, which leads to a continuous constituent. Then the subordinate clause would be added in a later derivation step. However, in the derivation for the best translation of SYS(1,1), this does not happen because a corresponding specific rule has not been learned. The translation produced by SYS(1,1) is not grammatical and misses important concepts, such as *geeignet* (*suitable*).



Source: er wäre damit auch geeignet gewesen , um die . . . zu fördern

Reference: it would thus be suitable to assist . . .

SYS(1,1): it would also have to be , in order to promote the . . .

SYS(2,1): he also would have been appropriate to promote the . . .

Figure 5: Test sentence with translations provided by the SCFG and the SLCFRS system, including the derivation of the SLCFRS system SYS(2,1).

6 Related Work

Several other translation models have been proposed which are expressive enough to generate the complex alignment configurations in Figure 1. Most notably, Galley and Manning (2010) propose a phrase-based translation system which allows for discontinuous phrase pairs, building upon the idea of a translation model proposed by Simard et al. (2005). They evaluate their system on a Chinese-to-English translation task and achieve some improvement in BLEU score over a phrase-based and a hierarchical phrase-based system. Unfortunately, we could not evaluate directly against their approach since the current documentation[7] of their system, Phrasal (Green et al., 2014), does not mention the discontinuous phrases anymore. We also could not obtain the data sets they used for their experiments.

In some sense, our work is the hierarchical, tree-based counterpart to the phrase-based approach of Galley and Manning (2010). This means that our translation grammar rules unify two types of "gaps" of previous approaches: (a) gaps in the sense of non-terminals that are inserted into longer phrases when hierarchical rules are created, as in Chiang (2007); their purpose is a better generalization of the translation rules, and (b) gaps in the sense of discontinuities in the yield of a translation rule, on the source side, on the target side or both, driven by the idea of allowing for more flexible phrases such that generated alignment structures are not restricted.

Besides the suggestion of Kaeshammer (2013) to use SLCFRS as the translation grammar formalism, which we have detailed and implemented in this work, Søgaard (2008) proposes to apply range concatenation grammar, an even more expressive formalism than LCFRS, and to use its ability to copy substrings during the derivation. This approach has downsides, such as no tight probabilities estimators, which are mentioned in Søgaard and Kuhn (2009).

An early advocate of translation modeling beyond context-free grammar formalisms is Melamed, who proposes to use Generalized Multitext Grammars, which are weakly equivalent to LCFRS (Melamed, 2004; Melamed et al., 2004). The incentive for this lies in linguistically motivated translation grammars and the general observation that discontinuous constituents are necessary for monolingual modelling of syntax.

7 Conclusions and Future Work

With this work, we extend the hierarchical phrase-based machine translation approach to discontinuous phrases, using SLCFRS as the translation grammar formalism. Since SLCFRS is a direct

[7] `http://www-nlp.stanford.edu/wiki/Software/Phrasal`, accessed on June 27, 2015

extension to SCFG, previous work on hierarchical phrase-based translation, in particular the model definition, training and decoding, can be extended to SLCFRS in a more or less direct manner. Evaluating our new system on a German-to-English translation task revealed a modest improvement in BLEU score over the SCFG baseline. Human evaluators showed a slight preference for translations produced by the SLCFRS system.

In the future, we will evaluate our approach on other language pairs, for example Chinese-English which has been used in related work. Furthermore, we would like to make use of recent advances in monolingual parsing of discontinuous constituents and use phrase-structure trees supporting discontinuous constituents for tree-based machine translation.

Acknowledgments

I would like to thank Laura Kallmeyer and Wolfgang Maier for discussions and comments and the reviewers for their suggestions. This research was funded by the German Research Foundation as part of the project *Grammar Formalisms beyond Context-Free Grammars and their use for Machine Learning Tasks*. Computational support and infrastructure was provided by the *Centre for Information and Media Technology* (ZIM) at the University of Düsseldorf (Germany).

References

Eberhard Bertsch and Mark-Jan Nederhof. 2001. On the complexity of some extensions of rcg parsing. In *IWPT*.

Pierre Boullier. 1998. Proposal for a Natural Language Processing syntactic backbone. Technical Report 3342, INRIA.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270.

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

Christian Federmann. 2012. Appraise: An open-source toolkit for manual evaluation of machine translation output. *The Prague Bulletin of Mathematical Linguistics*, 98:25–35, September.

Daniel Fernández-González and André Martins. 2015. Parsing as reduction. *arXiv preprint arXiv:1503.00030*.

Michel Galley and Christopher D. Manning. 2010. Accurate non-hierarchical phrase-based translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 966–974.

Spence Green, Daniel Cer, and Christopher D. Manning. 2014. Phrasal: A toolkit for new directions in statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 114–121. Association for Computational Linguistics.

Hieu Hoang and Philipp Koehn. 2010. Improved translation with source syntax labels. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 409–417. Association for Computational Linguistics.

Liang Huang and David Chiang. 2005. Better k-best parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 53–64. Association for Computational Linguistics.

Liang Huang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Annual Meeting-Association For Computational Linguistics*, volume 45, page 144.

Miriam Kaeshammer and Anika Westburg. 2014. On complex word alignment configurations. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1773–1780, Reykjavik, Iceland, May.

Miriam Kaeshammer. 2013. Synchronous linear context-free rewriting systems for machine translation. In *Proceedings of the Seventh Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 68–77. Association for Computational Linguistics.

Laura Kallmeyer and Wolfgang Maier. 2013. Data-driven parsing using Probabilistic Linear Context-Free Rewriting Systems. *Computational Linguistics*, 39(1).

Laura Kallmeyer. 2010. *Parsing beyond context-free grammars*. Springer Science & Business Media.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Wolfgang Maier and Timm Lichte. 2011. Characterizing discontinuity in constituent treebanks. In *Formal Grammer 2009, Revised Selected Papers*, volume 5591 of *LNAI*. Springer.

Wolfgang Maier. 2010. Direct parsing of discontinuous constituents in German. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*.

Wolfgang Maier. 2015. Discontinuous incremental shift-reduce parsing. In *Proceedings of ACL-IJCNLP 2015*, Beijing, China.

I. Dan Melamed, Giorgio Satta, and Benjamin Wellington. 2004. Generalized multitext grammars. In *Proceedings of the 42nd Annual Conference of the Association for Computational Linguistics (ACL)*.

I. Dan Melamed. 2004. Statistical machine translation by parsing. In *Proceedings of the 42nd Annual Conference of the Association for Computational Linguistics (ACL)*.

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

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, 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 *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Giorgio Satta and Enoch Peserico. 2005. Some computational complexity results for synchronous context-free grammars. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 803–810.

Michel Simard, Nicola Cancedda, Bruno Cavestro, Marc Dymetman, Eric Gaussier, Cyril Goutte, Kenji Yamada, Philippe Langlais, and Arne Mauser. 2005. Translating with non-contiguous phrases. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 755–762.

Anders Søgaard and Jonas Kuhn. 2009. Empirical lower bounds on alignment error rates in syntax-based machine translation. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation (SSST '09)*. Association for Computational Linguistics.

Anders Søgaard and Dekai Wu. 2009. Empirical lower bounds on translation unit error rate for the full class of inversion transduction grammars. In *Proceedings of the 11th International Conference on Parsing Technologies*, pages 33–36.

Anders Søgaard. 2008. Range concatenation grammars for translation. In *Proceedings of Coling 2008: Companion volume: Posters*.

Anders Søgaard. 2010. Can inversion transduction grammars generate hand alignments? In *Procceedings of the 14th Annual Conference of the European Association for Machine Translation (EAMT)*.

Andreas van Cranenburgh and Rens Bod. 2013. Discontinuous parsing with an efficient and accurate dop model. In *Proceedings of the International Conference on Parsing Technologies (IWPT 2013)*.

Yannick Versley. 2014. Experiments with easy-first nonprojective constituent parsing. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 39–53.

K. Vijay-Shanker, David Weir, and Aravind K. Joshi. 1987. Characterizing structural descriptions used by various formalisms. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*.

David Weir. 1988. *Characterizing Mildly Context-Sensitive Grammar Formalisms*. Ph.D. thesis, University of Pennsylviania, Philadelphia, PA.

Benjamin Wellington, Sonjia Waxmonsky, and I. Dan Melamed. 2006. Empirical lower bounds on the complexity of translational equivalence. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 977–984.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational linguistics*, 23(3):377–403.

Omar Zaidan. 2009. Z-mert: A fully configurable open source tool for minimum error rate training of machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 91:79–88.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of the Workshop on Statistical Machine Translation, HLT/NAACL*.