Bilingual Structured Language Models for Statistical Machine Translation

Ekaterina Garmash and **Christof Monz**
Informatics Institute, University of Amsterdam
Science Park 904, 1098 XH Amsterdam, The Netherlands
{e.garmash,c.monz}@uva.nl

Abstract

This paper describes a novel target-side syntactic language model for phrase-based statistical machine translation, bilingual structured language model. Our approach represents a new way to adapt structured language models (Chelba and Jelinek, 2000) to statistical machine translation, and a first attempt to adapt them to phrase-based statistical machine translation. We propose a number of variations of the bilingual structured language model and evaluate them in a series of rescoring experiments. Rescoring of 1000-best translation lists produces statistically significant improvements of up to 0.7 BLEU over a strong baseline for Chinese-English, but does not yield improvements for Arabic-English.

1 Introduction

Many model components of competitive statistical machine translation (SMT) systems are based on rather simplistic definitions with little linguistic grounding, which includes the definitions of phrase pairs, lexicalized reordering, and n-gram language models. However, earlier work has also shown that statistical MT can benefit from additional linguistically motivated models. Most prominent among the linguistically motivated approaches are syntax-based MT systems which take into account the syntactic structure of sentences through CKY decoding and categorial labels (Zollmann and Venugopal, 2006; Shen et al., 2008). On the other hand, the commonly used phrase-based SMT approaches can also reap some of the benefits of using syntactic information by integrating linguistic components addressing specific phenomena, such as Cherry (2008), Carpuat et al. (2010), Crego and Yvon (2010), Ge (2010), Xiang et al. (2011), Lerner and Petrov (2013), Garmash and Monz (2014).

This paper is a contribution to the existing body of work on how syntactically motivated models help translation performance. We work with the phrase-based SMT (PBSMT) (Koehn et al., 2003) framework as the baseline system. Our choice is motivated by the fact that PBSMT is a conceptually simple and therefore flexible framework. It is typically quite straightforward to integrate an additional model into the system. Also, PBSMT is the most widely used framework in the SMT research community, which ensures comparability of our results to other people's work on the topic.

There is a variety of ways syntax can be used in a PBSMT model. Typically a syntactic representation of a source sentence is used to define constraints on the order in which the decoder translates it. For example, Cherry (2008) defines soft constraints based on the notion of syntactic cohesion (Section 2). Ge (2010) captures reordering patterns by defining soft constraints based on the currently translated word's POS tag and the words structurally related to it. On the other hand, target syntax is more challenging to use in PBSMT, since a target-side syntactic model does not have access to the whole target sentence at decoding. Post and Gildea (2008) is one of the few target-side syntactic approaches applicable to PBSMT, but it has been shown not to improve translation. Their approach uses a target side parser as a language model: one of the reasons why it fails is that a parser assumes its input to be grammatical and chooses the most likely parse for it. What we are interested in during translation is how gram-

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 2398–2408,
Lisbon, Portugal, 17-21 September 2015. ©2015 Association for Computational Linguistics.

matical the target sentence actually is.

In addition to reordering constraints, source syntax can be used for target-side language modeling. A target side string can be encoded with source-syntactic building blocks and then scored as to how well-formed it is. Crego and Yvon (2010), Niehues et al. (2011), Garmash and Monz (2014) model target sequences as strings of tokens built from the target POS tag and the POS tags of the source words related to it through alignment and the source parse. In this paper, we define a target-side syntactic language model that takes structural constraints from the source sentence, but uses the words from the target side (as 'building blocks'). We do it by adapting an existing monolingual model of Chelba and Jelinek (2000), *structured language models*, to the bilingual setting. Our contributions can be summarized as follows:

- we propose a novel method to adapt monolingual structured language models (Chelba and Jelinek, 2000) (Section 3) to a PBSMT system (Section 4), which does not require an external on-the-fly parser, but only uses the given source-side syntactic analysis to infer structural relations between target words;
- building on the existing literature, we propose a set of deterministic rules that incrementally build up a parse of a target translation hypothesis based on the source parse (Section 4);
- we evaluate our models in a series of rescoring experiments and achieve statistically significant improvements of up to 0.7 BLEU for Chinese-English (Section 5).

Before describing the models, we motivate our method with a common assumption about cross-lingual correspondence (Section 2).

2 Direct correspondence assumption and syntactic cohesion in SMT

Before we apply the syntactic model introduced in Section 3 to the bilingual setting (Section 4), we first explain two widely used assumptions about syntactic correspondence across languages.

We take a *dependency tree* to be a syntactic representation of a sentence and reason about other syntactic assumptions and models in its terms. In this work, we choose a dependency structure over a constituency structure because the former is more primitive.[1] A dependency parse D is a dependency tree analysis of a sentence W, and we will think of it as a relation between words of W, such that $D(w, v)$ if w is a parent (head) of v (v being a child/modifier). D can be generalized to D^* which is an relation between words that are connected by a continuous path in a dependency tree (i.e. $D^*(w, v)$ if $D(w, v)$ or if $\exists u$ s.t. $D(w, u) \wedge D^*(u, v)$). We assume *unlabeled* dependency trees. Finally, we make a projectivity assumption, which is supported by empirical data in many languages (Kuhlmann and Nivre, 2006; Havelka, 2007), and makes a model computationally less expensive. A dependency parse D of a sentence $W = w_1, \ldots, w_n$ is *projective*, if for every word pair $w_i, w_j \in W$ s.t. $D(w_i, w_j)$ it holds that every $w_k \in W$ s.t. $i < k < j$ or $j < k < i$ is a descendant of w_i, i.e., $D^*(w_i, w_k)$; see Figure 1.

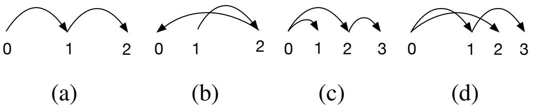


Figure 1: Examples of projective and non-projective parses. (a-b): projective (a) and non-projective (b) parses of the same dependency tree. (b) is non-projective because node 1 is not a descendant of either 0 or 2 (it is the parent of 2). (c-d): projective (c) and non-projective (d) parses of the same dependency tree. Node 2 in (d) is placed between its sibling (node 1) and the child of its sibling (node 3), neither of which is its ancestor.

Most NLP models that address the interaction of two or more languages are based (explicitly or implicitly) on the *direct correspondence assumption* (DCA) (Hwa et al., 2002). It states that close translation equivalents in different languages have the same dependency structure. This is grounded linguistically, as translation equivalence implies semantic equivalence and therefore thematic relations are preserved (Hwa et al., 2002). Thus dependency relations are preserved, as they are defined based on thematic relations between words. On the other hand, there is plenty empirical evidence supporting the violation of DCA under certain conditions (Hwa et al., 2002). For instance, even semantically very close sentences in different languages may have a different number of

[1] A dependency parse (a dependency tree analysis of a sentence) is more primitive because every constituency parse can be formalized as a projective dependency parse with labeled relations, but not vice versa (Osborne, 2008).

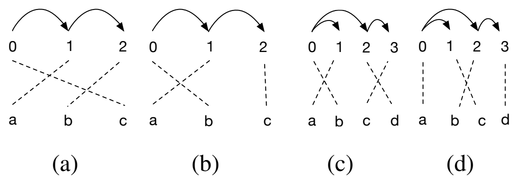


Figure 2: Examples of cohesive and uncohesive translations. (a-b): cohesive (a) and uncohesive (b) translations of the same dependency parse. (b) is uncohesive because words a and c translate the source subtree $\{(1,2)\}$, but the target word b does not translate this subtree. (c-d): cohesive (c) and uncohesive (d) translations. (d) is uncohesive because a and c translate the source subtree $\{(0,1)\}$, but b does not translate it.

words. Syntactic divergence increases if the two languages are typologically different.

Even though DCA only holds up to a certain level of precision, it is widely used in NLP. There are models of cross-lingual transfer that define syntactic structure of one language by conditioning it on the structure of semantically equivalent sentences in another language (Naseem et al., 2012). DCA has also been used in SMT. In particular, syntax-based SMT is built implicitly around this assumption (Wu, 1997; Yamada and Knight, 2001). In Quirk and Menezes (2006) DCA is explicitly implemented by defining a translation model in terms of treelet pairs where target-side treelets are produced by projecting source dependencies via word alignments.

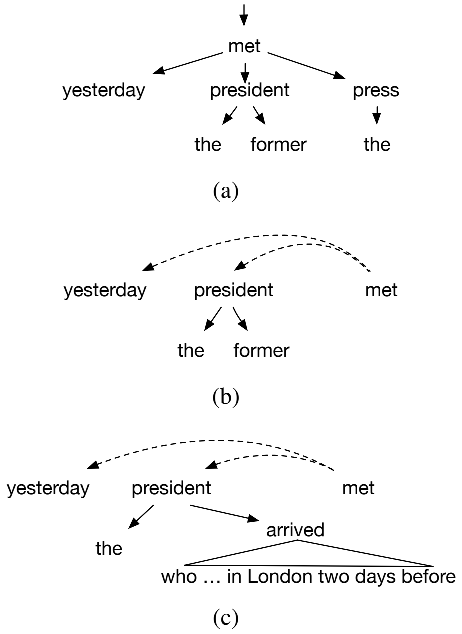
Closely related to DCA is the notion of *syntactic cohesion* of translation (Fox, 2002; Cherry, 2008). This is a constraint that does not allow for non-projective reordering: Given a source parse D_S, a translation W is cohesive if all translated target words w_i, w_j do not have any word w_k between them such that there is a source subtree sub in D_S such that some parts of it are translated by w_i and w_j but not by w_k (Figure 2). Cherry (2008) and Bach et al. (2009) define a set of soft constraints based on the syntactic cohesion assumption which are applicable to PBSMT decoding. They only require phrase applications, and not necessarily individual target words, to conform to the cohesion principle. For example, if we imagine a situation where a subtree as in Figure 2(b) is translated as a whole with one phrase application (and not word by word), then it does not violate the cohesion principle, although it is internally uncohesive. Both our approach and Cherry (2008) implement the idea of conforming the target translation to the source syntactic structure, but in different ways. Approaches like Cherry (2008) define principles that constrain the decoder in order to produce better translations. Our goal is to have a model that allows for a more direct way of evaluation of how well-formed the target translation is. In Section 5 we compare translation performance of the two approaches.

3 Structured language models

As discussed in Sections 1 and 2, we would like to test how much a PBSMT can benefit from an additional syntax-based LM. In this section, we describe a syntactic language model, structured LM (SLM) (Chelba and Jelinek, 2000), that we extend to a bilingual setting and apply to SMT in Section 4. SLMs have been applied in SMT before (Yamada and Knight, 2001; Yu et al., 2014), but as we show in Section 4, we provide a much simpler method to integrate it into the system. While a SLM is not the only syntactically defined LM, it is one of the few that models sentence generation sequentially. And due to the way the decoding procedure of PBSMT is defined, it is natural and straightforward to use models whose score can be computed sequentially. Other syntactic language models define sentence generation hierarchically (Shen et al., 2008; Sennrich, 2015), which complicates their integration into a PBSMT system.

The linguistic intuition behind SLMs is that the structural children of a word do not essentially change its distributional properties but just provide additional specification. In Figure 3(a) the word *president* has two modifiers: *the* and *former* and it follows *yesterday* (an adjunct) and precedes *met* (a predicate). This ordering is correct in English. If instead its modifier was *a* or an entire relative clause, it would not make it incorrect.

To capture this observation, (Chelba and Jelinek, 2000) propose a language model where each word w_i of a sentence W is predicted by an ordered subset of the words preceding w_i. This conditioning subset is selected based on the syntactic properties of the preceding sequence W_{i-1}: the strong predictors are kept and the weak ones are left out. The strong predictors are the set of *exposed heads*. Given a subsequence W_{i-1} and its associated parse D_{i-1}, exposed heads are the roots of all the disconnected subtrees in D_{i-1}. Note that



(a)

(b)

(c)

Figure 3: A fully parsed sentence (a) and its partial parse (b) during sequential generation. The partial parse in (b) has two disconnected subtrees with roots *yesterday* and *president*. These roots are the exposed heads for *met*. (c) is an alternative sentence with a similar structure: *president* is still a root of a subtree, and thus and an exposed head.

a parse D_{i-1} is not necessarily fully connected and thus a word can have multiple conditioning words.

For an example, consider again Figure 3(a). In a left-to-right scenario, when *met* is generated, a regular n-gram LM conditions it on *yesterday the former president*, while a SLM conditions it on *yesterday president*, since these two words are the exposed heads with respect to *met* (Figure 3(b)). The words *the* and *former* are modifiers of *president* and they get filtered out. Thus we obtain a less specific conditioning history, which may lead to the resulting model being less sparse. Another potential benefit is that SLMs can capture long-distance reordering: If *president* had as its modifier a relative clause (Figure 3(c)) then a simple n-gram LM would be conditioned on *days before* (assuming $n = 3$), while an SLM would condition *met* on *yesterday president*.

Summarizing the ideas of words being conditioned on a structurally defined subset of the preceding sentence, Chelba and Jelinek (2000) formalize the generation process of W as follows:[2] Each new word w_i is conditioned on a sequence of exposed heads $\mathit{Expos}(W, D)$. Then a tag t_i is predicted, and the parse D_{i-i} of W_{i-1} is extended to D_i incorporating w_i and t_i (where W_{i-1} is the prefix of W preceding w_i):

$$\begin{aligned} p(W,D) = \prod_{i=1}^{|W|} & p(w_i|\mathit{Expos}(W_{i-1}, D_{i-1})) \\ & \cdot p(t_i|w_i, \mathit{Expos}(W_{i-1}, D_{i-1})) \\ & \cdot p(D_i|w_i, t_i, \mathit{Expos}(W_{i-1}, D_{i-1})). \end{aligned} \tag{1}$$

They use a shift-reduce parser with reduce-left, reduce-right, and shift operations.

4 Bilingual structured language models

In this section, we combine the direct correspondence assumption (Section 2) and SLMs (Section 3), and define bilingual structured language models (BiSLMs) for PBSMT. Structured LMs have been successfully applied in SMT before. Yamada and Knight (2001) use SLMs in a string-to-tree SMT system where a derivation of a target-side parse tree is part of the decoding algorithm, and target syntactic representations are obtained 'for free'. Yu et al. (2014) use an on-the-fly shift-reduce parser to build an incremental target parse.

The approaches sketched above rely on resources that a standard PBSMT system does not have access to by default. Phrase-based decoders do not provide us with a parse of the target sentence, and inferring the parse of a target string with an external parser is computationally expensive and potentially unreliable (see Section 1). Our main insight is that in a bilingual setting one does not need an additional probabilistic target parsing model. We assume that the source parse is given (precomputed) and that the DCA (Section 2) holds, and project the parse deterministically onto the target side via word alignments[3]. We obtain the following equation:

$$\begin{aligned} p(T|S, D_S) = \prod_{i=1}^{|T|} & p(t_i|\, \mathit{Expos}(T_{i-1}, \\ & \mathit{ProjP}(D_S, S, T_{i-1}))), \end{aligned} \tag{2}$$

where T is a target sentence, T_{i-1} is the sequence in T preceding the i-th target word t_i, S is a

[2] The original model by (Chelba and Jelinek, 2000) is defined in terms of a lexicalized constituency grammar, but as we discussed in Section 2, constituency parses can be transformed into dependency parses.

[3] Phrase-internal word alignments are stored in the phrase table and are available at decoding time, see Section 4.4.

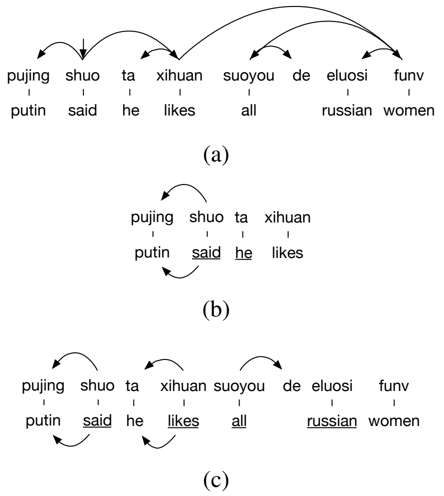

pujing shuo ta xihuan suoyou de eluosi funv
putin said he likes all russian women
(a)
pujing shuo ta xihuan
putin said he likes
(b)
pujing shuo ta xihuan suoyou de eluosi funv
putin said he likes all russian women
(c)

Figure 4: Chinese-English sentence pair (a) and sets of exposed heads (underlined) at different generation (b and c) steps of a bilingual SLM.

source sentence, D_S is a source dependency parse, and *ProjP* is a function that returns a partial target parse D_{Ti-1} by projecting D_S onto T_{i-1}. In words, at each time step i we predict the next word t_i conditioned on the exposed heads of the partial parse of T_{i-1} projected from the source side. We limit *Expos* to returning the four preceding exposed heads.[4] Because the function *ProjP* is deterministic and because we do not have to predict tags for words, Equation 2 is simpler than Equation 1.

We first illustrate Equation 2 with an example in Figure 4. Since word alignment is monotonic in Figure 4(a), it is straightforward to project the source dependencies onto the target side. We aim to imitate a monolingual parser in the way we build up our projected parse: Reduce operations should be invoked whenever both of the subtrees involved in the operation are complete, i.e., are not expected to have any more modifiers (Section 4.2). For example, when the target word *likes* is produced its exposed heads are *said* and *he* (Figure 4(b)), since *Putin* is a modifier of *said.* Likewise, the exposed heads for *women* are *said likes all Russian* (Figure 4(c)).

In what follows we discuss how to define *ProjP*. Compared to projection approaches like (Quirk and Menezes, 2006), we would like our model to project a source parse incrementally, allowing it to be used in a PBSMT decoder. We think of *ProjP* as a function that computes the output in two stages: first, it infers from the source parse the dependency relations between target words (Section 4.1), second, it decides how to parse the target sequence, i.e. in which order to assign these dependencies (Section 4.2). Additionally, in Section 4.3 we propose to use additional labelings of target words, and in Section 4.4 we describe some important implementation details.

4.1 Dependency graph projection

Adoption of DCA (Section 2) allows to build up a target dependency tree from a source tree by projecting the latter through word alignments. The definition of DCA can be rephrased as requiring a one-to-one correspondence *map* between words of a sentence pair, allowing one to unambiguously map dependencies: Given a source parse, if t_1 is the head of t_2, then $\mathit{map}(t_1)$ is the head of $\mathit{map}(t_2)$. The correspondence relation that we have in PBSMT is the word alignment *align*: in the most general case, it is a many-to-many correspondence, and the straightforward projection described above can lead to incorrect dependency structures. To overcome these problems, we describe a simple ordered set of projection rules, based on the ones specified by (Quirk and Menezes, 2006) (and we point out if otherwise).

The general idea behind this set of rules is to extract a one-to-one function ***align***$_{1-1}$ from source words to target words from *align* and use it to project source dependencies as described in the paragraph above (**R1** below). We then use additional rules (**R2-R4** below) for the target words that are not in *align*$_{1-1}$. Given a source sentence S with a parse D_S, a target sentence T and word alignment *align*, ***align***$_{1-1}$ is extracted as follows: For all $t_i \in T$ with multiple aligned source words $\{s_{i_1}, s_{i_2}, ...\}$ only *align*$_{1-1}(s_{i_1}) = t_i$ (only leftmost source word is kept, the links from the rest of the source words are removed[5]). For all $s_i \in S$ with aligned target words $\{t_{i_1}, t_{i_2}, ...\}$ keep the link only for the leftmost aligned target word: *align*$_{1-1}(s_i) = t_{i_1}$. For example, in Figure 5(b) the link between f_0 and e_1 is not in *align*$_{1-1}$, and in Figure 5(c) the link between f_1 and e_0 is removed (and the arc from f_2 to f_1 is not projected).

[4] As written above, we choose the dependency structures over the lexicalized constituency ones because the latter can be mapped to the former. It is thus more likely that a projected dependency tree is still be a well-formed parse, than a projected constituency tree. We decided to work with structural models that are more flexible, but one may also define BiSLM in terms of the more constraining constituency trees and see if the such model has better generalization power.

[5] This is an ad-hoc solution, other heuristics could be used.

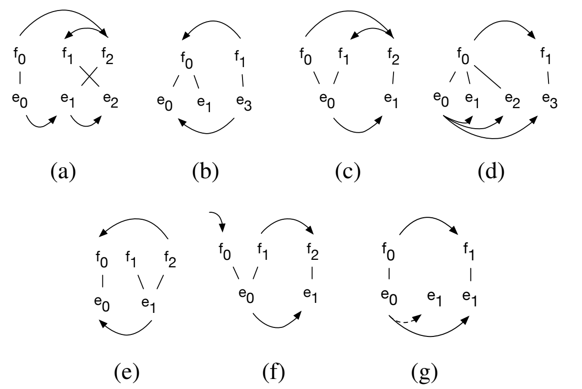


Figure 5: Examples for dependency projection rules. (a): no alignment links get removed (R1). (b): $f_0 - e_1$ link is removed from *align*$_{1-1}$ (R1). (c): $f_1 - e_0$ link gets removed (R1). (d): e_1 and e_2 get adjoined to e_0 (R2). (e): R3a. (f): R3b. (g) demonstrates two versions of R4: the dashed arrow gets 'realized' only if we adjoin unaligned words to the preceding head.

The following rules should be applied in order (as *else-if* conditions). Given a source sentence S with a parse D_S, a target sentence T and word alignment *align* between them, $t_i \in T$ is a head of $t_j \in T$ (i.e. $D_T(t_i, t_j)$):

(R1) if there are $s_k, s_l \in S$ s.t. $D_S(s_k, s_l)$ and *align*$_{1-1}(s_k) = t_i$ and *align*$_{1-1}(s_l) = t_j$; see Figures 5(a)-5(c);

(R2) if $\exists s \in S$ s.t. *align*$_{1-1}(s) = t_i$ and $(s, t_j) \in$ *align*. This rule deals with one-to-many alignments; see Figure 5(d);

(R3a) if $\exists s_k$ s.t. *align*$_{1-1}(s_k) = t_i$ and $\exists s_l$ s.t. $(s_l, t_j) \in$ *align* and and $D_S(s_l, s_k)$, and t_i linearly precedes t_j. In words: if two target words are in *align*$_{1-1}$ but do not get connected via **R1**, find a source word aligned to the second target word that may get them connected; see Figure 5(e);

(R3b) same as R3a, but in case t_j precedes t_i (i.e., find an additional source word aligned to the first target word; see Figure 5(f)).[6]

(R4) In case $\neg\exists s\ (s, t_j) \in$ *align* (t_j is unaligned), we consider two strategies: We simplify the rule of Quirk and Menezes (2006) (dealing with the same situation) by adjoining it to the immediately preceding head. We also consider a strategy whereby the word remains unconnected to any word in the sentence; see Figure 5(g).

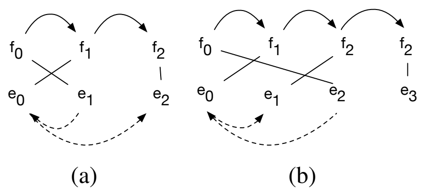


Figure 6: (a): The dashed lines are the dependency arcs that would project through word alignment, resulting in a non-projective projective (impossible under strong source-completeness). (b): The dashed lines are the parse produced under weak source-completeness. Under strong completeness none of the words will get connected.

4.2 BiSLM parsing procedure

Given an inference procedure for dependency relations between target words (Section 4.1), one can specify in which order the corresponding dependency arcs are assigned to the target sentence. We define an incremental parsing procedure in terms of three operations: *shift*, *left-reduce*, and *right-reduce*. The operations are applied as soon as the sufficient conditions hold: We specify the conditions using the following structural properties. A target subtree is *source-complete* if all the descendants of *align*$^{-1}_{1-1}$(*root(sub)*) (source correspondent of the root of the current subtree) (Section 4.1) have been translated and reduced. A target subtree is *complete* if it is source-complete and all the target words that are its children through non-projected arcs (through R2 or R4 in Section 4.1) have been translated and reduced. The bilingual parsing operations and the sufficient conditions for them are defined as follows:

Shift: after the word is produced it is shifted onto the stack as an elementary subtree.

Left-reduce: if a disconnected subtree *sub*$_i$ and a disconnected subtree *sub*$_{i-1}$ immediately preceding it are both complete and D_T(*root(sub*$_i$*)*, *root(sub*$_{i-1}$*)*), adjoin *sub*$_{i-1}$ to *sub*$_i$ so that *root(sub*$_{i-1}$*)* is a modifier of *root(sub*$_i$*)*.

Right-reduce: analogous to *left-reduce*, but D_T(*root(sub*$_{i-1}$*)*, *root(sub*$_i$*)*).

In the case of non-cohesive translation the resulting target dependencies are non-projective. Our definition of left- and right-reduce only produces projective parses. For a non-cohesive translation, certain subtrees will never be source-complete and will never be reduced; see Figure 6(a). Note that this is not a disadvantage

[6] R3a and R3b differ from the rules proposed in Quirk and Menezes (2006) dealing with the same situation, since we had to adapt it to the left-to-right parsing scenario.

of our model. Cherry (2008) simply assumes that non-cohesive reordering should be penalized, and our model is able to learn this pattern. We also consider an alternative to incorporating non-cohesive alignments by relaxing the definition of completeness for subtrees: A projected subtree sub is *weakly source-complete* if all descendants of all source word(s) which are aligned to the root of sub have been translated and, *only* if the definition of reduce applies, reduced; see Figure 6(b).

4.3 Syntactic labeling of tokens

One of the problems with SLMs in general is that at time steps i and j the sets of exposed heads for t_i and t_j can differ in size, which may imply different predictive power. To this end, we add an additional detail to our model: Each time a reduction occurs, we label the root of the subtree to which another subtree has been adjoined, thus making the conditioning history more specific. We use the following labelings:

Reduction labeling: if a subtree is adjoint to *sub* from the left, then label *root(sub)* with **LR**. If it is adjoint from the right, then label it with **RR**.

Reduction POS-labeling: same as in simple reduction labeling, but add the POS tag of the root of the reduced subtree to the label.

4.4 Implementation and training

To use BiSLM during decoding, one needs access to phrase-internal alignments and target POS tags. We store phrase-internal alignments and target-side POS annotations of each phrase in the phrase table, based on the most frequent internal alignment during training and the most likely target-side POS labeling $\hat{t}$ given the phrase pair: $\hat{t} = \arg\max_{\bar{t}} p(\bar{t}|\bar{e}, \bar{f})$. We train BiSLMs on the parallel training data (Section 5.1) and use the Stanford dependency parser (Chang et al., 2009) for Chinese and and the Stanford constituency parser (Green and Manning, 2010) for Arabic[7]. POS-tagging of the training data is produced with the Stanford POS-tagger (Toutanova et al., 2003). We learn a 5-gram model using SRILM (Stolcke et al., 2011) with modified Kneser-Ney smoothing.

5 Experiments

To evaluate the effectiveness of BiSLMs for PB-SMT, we performed rescoring experiments for Arabic-English and Chinese-English. We compare the resulting 1-best translation lists with an output of the baseline system and the baseline augmented with soft cohesion constraints from Bach et al. (2009).

System	MT06	MT08	MT06+MT08
baseline	32.60	25.94	29.56
cohesion	32.52	25.98	29.54

Table 1: Chinese-English baseline and comparison model (Cherry, 2008; Bach et al., 2009) results.

System	MT08	MT09	MT08+MT09
baseline	45.84	48.61	47.18
cohesion constr.	45.61	48.49	47.02

Table 2: Arabic-English baseline and comparison model (Cherry, 2008; Bach et al., 2009) results.

5.1 Experimental setup

This section provides information about our baseline system. Word-alignment is produced with GIZA++ (Och and Ney, 2003). We use an in-house implementation of a PBSMT system similar to Moses (Koehn et al., 2007). Our baseline has all standard PBSMT features including language model, lexical weighting, and lexicalized reordering. The distortion limit is set to 5. A 5-gram LM is trained on the English Gigaword corpus (1.6B tokens) using SRILM with modified Kneser-Ney smoothing and linear interpolation. Information about the training data for the Arabic-English and Chinese-English systems is in Table 3.[8] Feature weights are tuned using pairwise ranking optimization (Hopkins and May, 2011) on the MT04 benchmark (for both language pairs). For testing, we use MT08 and MT09 for Arabic, and MT06 and MT08 for Chinese. We use case-insensitive BLEU (Papineni et al., 2002) as evaluation metric. Approximate randomization (Noreen, 1989; Riezler and Maxwell, 2005) is used to detect statistically significant differences.

5.2 Baseline and comparison systems

As a comparison model, we implemented six features from Cherry (2008) and Bach et al. (2009)[9] and added them to the log-linear interpolation used

[7] We extract dependency parses from its output based on Collins (1999)

[8] The standard LDC corpora were used for training.

[9] Exhaustive and non-exhaustive interruption check, exhaustive and non-exhaustive interruption count, verb- and noun-dominated subtree interruption count.

Training set	N. of lines	N. of tokens
Source side of Ar-En set	4,376,320	148M
Target side of Ar-En set	4,376,320	146M
Source side of Ch-En set	2,104,652	20M
Target side of Ch-En set	2,104,652	28M

Table 3: Training data for Arabic-English and Chinese-English experiments.

by the baseline system. Since these features are binary or count-based, we cannot use them directly in rescoring. For that reason we integrated the features into the decoder and tuned the corresponding weights. The results for Chinese-English and Arabic-English translation experiments are presented in Table 1 and 2, respectively. We see that adding the cohesion constraints does not improve performance. This finding is different from, for example, Feng et al. (2010), where they get improvement for Chinese-English: however, we note that their training set is smaller than ours, and their baseline is weaker as it does not contain lexicalized distortion models.

5.3 Rescoring experiments

Rescoring with BiSLMs is performed as follows: For the test runs of the baseline system we compute the $n = 1000$ best translation hypotheses for each source sentence and extract their derivations (sequence of phrase pair applications). Each phrase pair in our implementation is associated with a unique phrase-internal alignment and target POS-sequence. We fully reconstruct word-alignment for each pair of a source sentence and its translation hypothesis. We project a precomputed source parse onto the target side and compute representations of the target sentence to be computed by a BiSLM. For each hypothesis, we take its BiSLM score and its score assigned by the baseline system and compute the final score as a weighted sum of the original baseline score and a length-normalized BiSLM score[10], where the weight λ is empirically set to 0.3:

$$\lambda \cdot \frac{score_{\text{BiSLM}}}{length_{\text{Hypothesis}}} + (1 - \lambda) \cdot score_{\text{Baseline}} \quad (3)$$

5.3.1 Chinese-English

Our main focus here is Chinese-English, since it has more instances of longer-distance reordering, at which syntax-based models are typically good. SLMs by design are good at capturing longer-distance dependencies. We try out several variations of BiSLM. First, we test whether to use a strong or weak definition of a complete subtree (Section 4.2). Second, we investigate whether to adjoin unaligned target words to a preceding head (Section 4.1; **unalign-adjoin+/-**). Third, we compare several target-side labeling methods (Section 4.3): plain (just target words), reduce (**LR** or **RR**) or reduce-POS (**LR**_POS or **RR**_POS, where POS is the tag of the root of the reduced subtree). The rescoring results are presented in Table 4.

labeling	**complete**	**unalign -adjoin**	BLEU	diff.
plain	strong	+	30.09▲	+0.53
		-	30.20▲	+0.64
	weak	+	30.11▲	+0.55
		-	30.22▲	+0.66
reduce	strong	+	29.94△	+0.40
		-	30.19▲	+0.63
	weak	+	30.09▲	+0.53
		-	30.24▲	+0.68
reduce-POS	strong	+	30.09▲	+0.53
		-	30.25▲	+0.69
	weak	+	30.05▲	+0.49
		-	30.25▲	+0.69

Table 4: Rescoring experiments for Chinese MT06+08 1000-best translation sets. Unrescored BLEU is 29.56. The column **labeling** contains information about the kind of labeling used on the target side of a BiSLM: just target words, target words with a reduction label, or target words with a reduction label and a POS of the root of the reduced subtree (Section 4.3). The column **complete** indicates whether we use a strong or weak definition of a complete subtree (Section 4.2). The column **unalign-adjoin** indicates whether we adjoin an unaligned target word to the preceding subtree (Section 4.1). Statistically significant improvements over the baseline are marked ▲ at the $p < .01$ level and △ at the $p < .05$ level. ▼ marks significant decrease at the $p < .01$ level.

The results show statistically significant improvement over the baseline of up to 0.7 BLEU (for all of the employed BiSLM variants except one). The rescoring experiments also demonstrate the tendency of the **unalign-adjoin-** feature value to produce higher scores than **unalign-adjoin+**. But the other two distinguishing features do not have an effect on BLEU scores. As future work, we are interested in examining if these features produce the same distribution of scores when a BiSLM is fully integrated into the decoder.

[10] Normalization is needed to ensure comparability of scores for translation hypotheses of different lengths, since longer translation hypotheses will have lower scores.

labeling	**complete**	**unalign -adjoin**	BLEU	diff.
plain	strong	+	47.20	-0.02
		-	$47.00^{\blacktriangledown}$	-0.18
	weak	+	47.22	+0.04
		-	$46.98^{\blacktriangledown}$	-0.20
reduce	strong	+	47.15	-0.03
		-	$46.99^{\blacktriangledown}$	-0.19
	weak	+	47.09	-0.09
		-	$46.98^{\blacktriangledown}$	-0.20
reduce-POS	strong	+	47.15	-0.03
		-	$46.98^{\blacktriangledown}$	-0.20
	weak	+	47.17	+0.01
		-	$47.00^{\blacktriangledown}$	-0.18

Table 5: Rescoring experiments for Arabic MT08+09 n-best translation sets. Unrescored BLEU for is 47.18. For notation see Table 4.

5.3.2 Arabic-English

We also rescore the n-best lists for the output of the Arabic-English baseline system and results are shown in Table 5. Arabic and English are typologically very different, but the range of reordering is much smaller than for Chinese-English. We expect reordering-related models to have lesser effect on Arabic as compared to Chinese (Carpuat et al., 2010). Experimental results on Arabic-English could indicate what kind of translation aspect benefits from BiSLMs. We see that for Arabic-English, just as for the cohesion constraint, BiSLM have little effect on BLEU scores, or even decrease them. This is a weak indication that BiSLMs are better at capturing reordering aspects. As for the varying features defining different BiSLM versions, we again see little effect of the labeling type or subtree completeness definition. On the other hand, we see the opposite pattern for the **unalign-adjoin** feature, where **unalign-adjoin+** is preferred.

To gain further insight into the different effect of BiSLM on the two language pairs, we evaluated our experimental output against a reordering-sensitive metric LRscore (Birch et al., 2010). We use the version of LRscore which is an average of the inverse Kendall's Tau distance and the Hamming distance. In order to compute alignments for test sets which are needed to compute the score we concatenated the parallel text with an additional 250K lines of parallel text from the training data to ensure better generalization of the alignment algorithm (GIZA++). The LRscores of the baseline are compared to the best performing BiSLM system with respect to BLEU, for each of the language pair. The results are provided in Tables 6 and 7.

system	LRscore MT06+08
baseline	0.4736
BiSLM	0.4907

Table 6: LRscores (average inverse Kendall's Tau distance and Hamming distance) for Chinese-English baseline and BiSLM with reduce-labeling, weak completeness, unalign-adjoin-.

system	LRscore MT08+09
baseline	0.6671
BiSLM	0.6719

Table 7: LRscores for Arabic-English baseline and BiSLM with plain-labeling, weak completeness, unalign-adjoin+.

As expected, the scores for Chinese-English are much lower than for Arabic-English, which is consistent with the observation reordering is more difficult for Chinese-English. BiSLM yields larger improvements for Chinese-English suggesting that the proposed model helps addressing difficult reordering problems. While there are also small improvements for Arabic-English the they may be too small to be detectable by BLEU.

6 Conclusions

In this paper we proposed a novel way to adapt structured language models to phrase-based SMT. Our method requires minimal changes to the PBSMT pipeline. We tried a number of variations of our model and evaluated them in rescoring experiments, resulting in statistically significant improvement for Chinese-English. The model is based on the idea of syntactic transfer (DCA; Section 2) and the positive result indicates its ability to capture syntactic patterns across languages. For Arabic-English, we did not observe any improvements, suggesting that our models indeed mainly improve reordering aspects. Improvements in rescoring are a positive indication that our model may be a strong feature during decoding. As future work, we will fully integrate our model into a PBSMT decoder and evaluate it on other language pairs with different reordering distributions.

Acknowledgments

We thank the reviewers for their useful comments. This research was funded in part by the Netherlands Organization for Scientific Research (NWO) under project numbers 639.022.213 and 612.001.218.

References

Nguyen Bach, Stephan Vogel, and Colin Cherry. 2009. Cohesive constraints in a beam search phrase-based decoder. In *Proceedings of the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1–4. Association for Computational Linguistics.

Alexandra Birch, Miles Osborne, and Phil Blunsom. 2010. Metrics for mt evaluation: evaluating reordering. *Machine Translation*, 24(1):15–26.

Marine Carpuat, Yuval Marton, and Nizar Habash. 2010. Improving Arabic-to-English statistical machine translation by reordering post-verbal subjects for alignment. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 178–183. Association for Computational Linguistics.

Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D. Manning. 2009. Discriminative reordering with chinese grammatical relations features. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*, pages 51–59. Association for Computational Linguistics.

Ciprian Chelba and Frederick Jelinek. 2000. Structured language modeling. *Computer Speech and Language*, 14(4):283–332.

Colin Cherry. 2008. Cohesive phrase-based decoding for statistical machine translation. In *Proceedings of Association for Computational Linguistics*, pages 72–80.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Josep M. Crego and François Yvon. 2010. Improving reordering with linguistically informed bilingual n-grams. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 197–205. Association for Computational Linguistics.

Minwei Feng, Arne Mauser, and Hermann Ney. 2010. A source-side decoding sequence model for statistical machine translation. In *Conference of the Association for Machine Translation in the Americas, Denver, Colorado, USA*.

Heidi J. Fox. 2002. Phrasal cohesion and statistical machine translation. In *Proceedings the Conference on Empirical Methods in Natural Language Processing*, pages 304–311. Association for Computational Linguistics.

Ekaterina Garmash and Christof Monz. 2014. Dependency-based bilingual language models for reordering in statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1689–1700, Doha, Qatar, October. Association for Computational Linguistics.

Niyu Ge. 2010. A direct syntax-driven reordering model for phrase-based machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 849–857. Association for Computational Linguistics.

Spence Green and Christopher D. Manning. 2010. Better arabic parsing: Baselines, evaluations, and analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 394–402. Association for Computational Linguistics.

Jiri Havelka. 2007. Beyond projectivity: Multilingual evaluation of constraints and measures on non-projective structures. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 608–615. Association for Computational Linguistics.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362. Association for Computational Linguistics.

Rebecca Hwa, Philip Resnik, Amy Weinberg, and Okan Kolak. 2002. Evaluating translational correspondence using annotation projection. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 392–399. Association for Computational Linguistics.

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 Association for Computational Linguistics on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics.

Marco Kuhlmann and Joakim Nivre. 2006. Mildly non-projective dependency structures. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 507–514, Sydney, Australia, July. Association for Computational Linguistics.

Uri Lerner and Slav Petrov. 2013. Source-side classifier preordering for machine translation. In *Proceedings of the Empirical Methods in Natural Language Processing*.

Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 629–637. Association for Computational Linguistics.

Jan Niehues, Teresa Herrmann, Stephan Vogel, and Alex Waibel. 2011. Wider context by using bilingual language models in machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 198–206. Association for Computational Linguistics.

Eric W. Noreen. 1989. *Computer Intensive Methods for Testing Hypotheses. An Introduction*. Wiley-Interscience.

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

Timothy Osborne. 2008. Major constituents and two dependency grammar constraints on sharing in coordination. *Linguistics*, 46(6):1109–1165.

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 of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

Matt Post and Daniel Gildea. 2008. Parsers as language models for statistical machine translation. In *Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas*, pages 172–181. Citeseer.

Chris Quirk and Arul Menezes. 2006. Dependency treelet translation: The convergence of statistical and example-based machine translation? *Machine Translation*, 20:43–65, March.

Stefan Riezler and John T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

Rico Sennrich. 2015. Modelling and optimizing on syntactic n-grams for statistical machine translation. *Transactions of the Association for Computational Linguistics*, 3:169–182.

Libin Shen, Jinxi Xu, and Ralph M. Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of the Association for Computational Linguistics*, pages 577–585.

Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash. 2011. Srilm at sixteen: Update and outlook. In *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop*, page 5.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 173–180. Association for Computational Linguistics.

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

Bing Xiang, Niyu Ge, and Abraham Ittycheriah. 2011. Improving reordering for statistical machine translation with smoothed priors and syntactic features. In *Proceedings of the Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 61–69. Association for Computational Linguistics.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 523–530. Association for Computational Linguistics.

Heng Yu, Haitao Mi, Liang Huang, and Qun Liu. 2014. A structured language model for incremental tree-to-string translation. *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics*, pages 1133–1143.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 138–141. Association for Computational Linguistics.