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-
matically 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 a 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. \( D(w, u) \land D^*(u, v) \)). 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.

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 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.](image-url)
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
the former president met yesterday

(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 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 $\text{Expos}(W, D)$. Then a tag $t_i$ is predicted, and the parse $D_{i-1}$ 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 $w_i$):

$$p(W, D) = \prod_{i=1}^{\left| W \right|} p(\text{Expos}(W_{i-1}, D_{i-1}), t_i | w_i) \\ 
\cdot p(t_i | w_i, \text{Expos}(W_{i-1}, D_{i-1})) \\
\cdot p(D_i | w_i, t_i, \text{Expos}(W_{i-1}, D_{i-1})). \tag{1}$$

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 SLs 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:

$$p(T | S, D_S) = \prod_{i=1}^{\left| T \right|} p(t_i | \text{Expos}(T_{i-1}, \text{ProjP}(D_S, S, T_{i-1})))) \tag{2}$$

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.
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 $Prov$ is a function that returns a partial target parse $D_{T_{i-1}}$ 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. Because the function $Prov$ 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 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 matches 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 $Prov$. 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 $Prov$ 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 $map(t_1)$ is the head of $map(t_2)$. 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 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. Given a source sentence $S$ with a parse $D_S$, a target sentence $T$ and word alignment $align$, $align_{i-1}$ is extracted as follows: For all $t_i \in T$ with multiple aligned source words $\{s_{i_1}, s_{i_2}, \ldots\}$ only $align_{i-1}(s_{i_1}) = t_i$ (only leftmost source word is kept, the links from the rest of the source words are removed). For all $s_i \in S$ with aligned target words $\{t_{i_1}, t_{i_2}, \ldots\}$ keep the link only for the leftmost aligned target word: $align_{i-1}(s_i) = t_{i_1}$. For example, in Figure 5(b) the link between $f_0$ and $e_1$ is not in align, 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).

---

4This is an ad-hoc solution, other heuristics could be used.
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 $\text{align}$ 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 $\text{align}_{1-1}(s_k) = t_i$ and $\text{align}_{1-1}(s_l) = t_j$; see Figures 5(a)-5(c);

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

(R3a) if $\exists s \in S$ and $\exists s_l \in S$ s.t. $(s, t_j) \in \text{align}$ and $D_S(s_l, s_k)$, and $t_i$ linearly precedes $t_j$. In words: if two target words are in $\text{align}_{1-1}$ 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)).

(R4) In case $\neg \exists s \ (s, t_j) \in \text{align} (t_j$ 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).

\footnote{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.}

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 $\text{align}_{1-1}^{-1}(\text{root}(sub))$ (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 $\text{sub}_i$ and a disconnected subtree $\text{sub}_{i-1}$ immediately preceding it are both complete and $D_T(\text{root}(\text{sub}_i), \text{root}(\text{sub}_{i-1}))$, adjoin $\text{sub}_{i-1}$ to $\text{sub}_i$ so that $\text{root}(\text{sub}_{i-1})$ is a modifier of $\text{root}(\text{sub}_i)$.

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

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.

---

Figure 5: Examples for dependency projection rules. (a): no alignment links get removed (R1). (b): $f_0 - e_1$ link is removed from $\text{align}_{1-1}$ (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.

---

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.
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 \textit{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:

\[
\hat{t} = \arg \max \bar{p}(\bar{e}, \bar{f})
\]

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 with modified Kneser-Ney smoothing.

### 5 Experiments

To evaluate the effectiveness of BiSLMs for PBSMT, 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).

<table>
<thead>
<tr>
<th>System</th>
<th>MT06</th>
<th>MT08</th>
<th>MT06+MT08</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>32.60</td>
<td>25.94</td>
<td>29.56</td>
</tr>
<tr>
<td>cohesion</td>
<td>32.52</td>
<td>25.98</td>
<td>29.54</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>System</th>
<th>MT08</th>
<th>MT09</th>
<th>MT08+MT09</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>45.84</td>
<td>48.61</td>
<td>47.18</td>
</tr>
<tr>
<td>cohesion constr.</td>
<td>45.61</td>
<td>48.49</td>
<td>47.02</td>
</tr>
</tbody>
</table>

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, exhausitive and non-exhaustive interruption count, verb- and noun-dominated subtree interruption count.
Table 3: Training data for Arabic-English and Chinese-English experiments.

<table>
<thead>
<tr>
<th>Source side of Ar-En set</th>
<th>N. of lines</th>
<th>N. of tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>4,376,320</td>
<td>148M</td>
<td></td>
</tr>
<tr>
<td>Target side of Ar-En set</td>
<td>4,376,320</td>
<td>146M</td>
</tr>
<tr>
<td>Source side of Ch-En set</td>
<td>2,104,652</td>
<td>28M</td>
</tr>
<tr>
<td>Target side of Ch-En set</td>
<td>2,104,652</td>
<td>20M</td>
</tr>
</tbody>
</table>

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.

$$\lambda \cdot \frac{\text{score}_{\text{BiSLM}}}{\text{length}_{\text{Hypothesis}}} + (1 - \lambda) \cdot \text{score}_{\text{Baseline}}$$

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 $\lambda$ is empirically set to 0.3:

---

\(^{10}\)Normalization is needed to ensure comparability of scores for translation hypotheses of different lengths, since longer translation hypotheses will have lower scores.
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 \textit{unalign-adjoin} feature, where \textit{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.

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 PB-SMT 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


Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency


