Dependency-Based Bilingual Language Models for
Reordering in 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 presents a novel approach to improve reordering in phrase-based machine translation by using richer, syntactic representations of units of bilingual language models (BiLMs). Our method to include syntactic information is simple in implementation and requires minimal changes in the decoding algorithm. The approach is evaluated in a series of Arabic-English and Chinese-English translation experiments. The best models demonstrate significant improvements in BLEU and TER over the phrase-based baseline, as well as over the lexicalized BiLM by Niehues et al. (2011). Further improvements of up to 0.45 BLEU for Arabic-English and up to 0.59 BLEU for Chinese-English are obtained by combining our dependency BiLM with a lexicalized BiLM. An improvement of 0.98 BLEU is obtained for Chinese-English in the setting of an increased distortion limit.

1 Introduction
In statistical machine translation (SMT) reordering (also called distortion) refers to the order in which source words are translated to generate the translation in the target language. Word orders can differ significantly across languages. For instance, Arabic declarative sentences can be verb-initial, while the corresponding English translation should realize the verb after the subject, hence requiring a reordering. Determining the correct reordering during decoding is a major challenge for SMT. This problem has received a lot of attention in the literature (see, e.g., Tillmann (2004), Zens and Ney (2003), Al-Onaizan and Papineni (2006)), as choosing the correct reordering improves readability of the translation and can have a substantial impact on translation quality (Birch, 2011). In this paper, we only consider those approaches that include a reordering feature function into the log-linear interpolation used during decoding.

The simplest reordering model is linear distortion (Koehn et al., 2003) which scores the distance between phrases translated at steps \( t \) and \( t + 1 \) of the derivation. This model ignores any contextual information, as the distance between translated phrases is its only parameter. Lexical distortion modeling (Tillmann, 2004) conditions reordering probabilities on the phrase pairs translated at the current and previous steps. Unlike linear distortion, it characterizes reordering not in terms of distance but type: monotone, swap, or discontinuous.

In this paper, we base our approach to reordering on bilingual language models (Marino et al., 2006; Niehues et al., 2011). Instead of directly characterizing reordering, they model sequences of elementary translation events as a Markov process. Originally, Marino et al. (2006) used this kind of model as the translation model, while more recently it has been used as an additional model in PBSMT systems (Niehues et al., 2011). We adopt and generalize the approach of Niehues et al. (2011) to investigate several variations of bilingual language models. Our method consists of labeling elementary translation events (tokens of bilingual LMs) with their different contextual properties.

What kind of contextual information should be incorporated in a reordering model? Lexical information has been used by Tillmann (2004) but is known to suffer from data sparsity (Galley and Manning, 2008). Also previous contributors to bilingual language modeling (Marino et al., 2006; Niehues et al., 2011) have mostly used lexical information, although Crego and Yvon (2010a) and Crego and Yvon (2010b) label bilingual to-

1Note that the standard PBSMT translation model assumes that events of translating separate phrases in a sentence are independent.
kens with a rich set of POS tags. But in general, reordering is considered to be a syntactic phenomenon and thus the relevant features are syntactic (Fox, 2002; Cherry, 2008). Syntactic information is incorporated in tree-based approaches in SMT, allowing one to provide a more detailed definition of translation events and to redefine decoding as parsing of a source string (Liu et al., 2006; Marton and Resnik, 2008), of a target string (Shen et al., 2008), or both (Chiang, 2007; Chiang, 2010). Reordering is a result of a given derivation, and CYK-based decoding used in tree-based approaches is more syntax-aware than the simple PBSMT decoding algorithm. Although tree-based approaches potentially offer a more accurate model of translation, they are also a lot more complex and requiring more intricate optimization and estimation techniques (Huang and Mi, 2010).

Our idea is to keep the simplicity of PBSMT but move towards the expressiveness typical of tree-based models. We incrementally build up the syntactic representation of a translation during decoding by adding precomputed fragments from the source parse tree. The idea to combine the merits of the two SMT paradigms has been proposed before, where Huang and Mi (2010) introduce incremental decoding for a tree-based model. On a very general level, our approach is similar to theirs in that it keeps track of whether the visited subtrees that are being translated at consecutive decoding steps. An important difference is that they keep track of whether the visited subtrees have been fully translated, while in our approach, once a syntactic structural unit has been added to the history, it is not updated anymore.

In this paper, we focus on source syntactic information. During decoding we have full access to the source sentence, which allows us to obtain a better syntactic analysis (than for a partial sentence) and to precompute the units that the model operates with. We investigate the following research questions: How well can we capture reordering regularities of a language pair by incorporating source syntactic parameters into the units of a bilingual language model? What kind of source syntactic parameters are necessary and sufficient?

Our contributions can be summarized as follows: We argue that the contextual information used in the original bilingual models (Niehues et al., 2011) is insufficient and introduce a simple model that exploits source-side syntax to improve reordering (Sections 2 and 3). We perform a thorough comparison between different variants of our general model and compare them to the original approach. We carry out translation experiments on multiple test sets, two language pairs (Arabic-English and Chinese-English), and with respect to two metrics (BLEU and TER). Finally, we present a preliminary analysis of the reorderingst resulting from the proposed models (Section 4).

2 Motivation

In this section, we elaborate on our research questions and provide background for our approach. We also discuss existing bilingual n-gram models and argue that they are often not expressive enough to differentiate between alternative reorderings. We should first note that the most commonly used n-gram model to distinguish between reorderings is a target language model, which does not take translation correspondence into account and just models target-side fluency. Al-Onaizan and Papineni (2006) show that target language models by themselves are not sufficient to correctly characterize reordering. In what follows we only discuss bilingual models.

The word-aligned sentence pair in Figure 1.a demonstrates a common Arabic-English reordering. As stated in the introduction, bilingual language models capture reordering regularities as a sequence of elementary translation events\(^3\). In the given example, one could decompose the sequential process of translation as follows: First translate the first word Alwzyr as the minister, then ArjE as attributed, then ArtfAE as the increase and so on. The sequence of elementary translation events is modeled as an n-gram model (Equation 1, where \(t_i\) is a translation event). There are numerous ways in which \(t_i\) can be defined. Below we first discuss how they have been defined within previous approaches, and then introduce our definition.

\[
p(t_1, \ldots, t_m) = \prod_{i=1}^{m} p(t_i|t_{i-n+1} \ldots t_{i-1})
\]

2.1 Lexicalized bilingual LMs

By including both source and target information into the representation of translation events we ob-

\(^2\)We used Buckwalter transliteration for Arabic words.

\(^3\)By an elementary translation event we mean a translation of some substructure of a sentence.
tain a bilingual LM. The richer representation allows for a finer distinction between reorderings. For example, Arabic has a morphological marker of definiteness on both nouns and adjectives. If we first translate a definite adjective and then an indefinite noun, it will probably not be a likely sequence according to the translation model. This kind of intuition underlies the model of Niehues et al. (2011), a bilingual LM (BiLM), which defines elementary translation events $t_1, \ldots, t_n$ as follows:

$$t_i = \langle e_i, \{ f \mid f \in A(e_i) \} \rangle,$$

where $e_i$ is the $i$-th target word and $A : E \rightarrow \mathcal{P}(F)$ is an alignment function, $E$ and $F$ referring to target and source sentences, and $\mathcal{P}(\cdot)$ is the powerset function. In other words, the $i$-th translation event consists of the $i$-th target word and all source words aligned to it. Niehues et al. (2011) refer to the defined translation events $t_i$ as bilingual tokens and we adopt this terminology.

There are alternative definitions of bilingual language models. Our choice of the above definition is supported by the fact that it produces an unambiguous segmentation of a parallel sentence into tokens. Ambiguous segmentation is undesirable because it increases the token vocabulary, and thus the model sparsity. Another disadvantage comes from the fact that we want to compare permutations of the same set of elements. For example, the two different segmentations of $ba$ into $[ba]$ and $[b][a]$ still represent the same permutation of the sequence $ab$. In Figure 1 one can produce a segmentation of $(AsEAr Albtrwl, oil prices)$ into $(Albtrwl, oil)$ and $(AsEAr, prices)$ or leave it as is. If we allow for both segmentations, the learnt probability parameters may be different for the sum of $(Albtrwl, oil)$ and $(AsEAr, prices)$ and for the unsegmented phrase.

Durrani et al. (2011) introduce an alternative method for unambiguous bilingual segmentation where tokens are defined as minimal phrases, called minimal translation units (MTUs). Figure 1 compares the BiLM and MTU tokenization for a specific example. Since Niehues et al. (2011) have shown their model to work successfully as an additional feature in combination with commonly used standard phrase-based features, we use their approach as the main point of reference and base our approach on their segmentation method. In the rest of the text we refer to Niehues et al. (2011) as the original BiLM. At the same time, we do not see any specific obstacles for combining our work with MTUs.

2.2 Suitability of lexicalized BiLM to model reordering

As mentioned in the introduction, lexical information is not very well-suited to capture reordering regularities. Consider Figure 2.a. The extracted sequence of bilingual tokens is produced by aligning source words with respect to target words (so that they are in the same order), as demonstrated by the shaded part of the picture. If we substituted the Arabic translation of Egyptian for the Arabic translation of Israeli, the reordering should remain the same. What matters for reordering is the syntactic role or context of a word. By using unnecessarily fine-grained categories we risk running into sparsity issues.

Niehues et al. (2011) also described an alternative variant of the original BiLM, where words are substituted by their POS tags (Figure 2.a, shaded part). Also, however, POS information by itself may be insufficiently expressive to separate cor-

---

(a) The original word alignment.

(b) BiLM tokens extracted from sentence (a).

(c) MTU tokens extracted from sentence (a).

Figure 1: Arabic-English parallel sentence, automatically word-aligned. The bilingual token sequences are produced according to two alternative definitions (BiLM and MTU).
Figure 2: Arabic-English parallel sentence, automatically parsed and word-aligned, with corresponding sequences of bilingual tokens (in the shaded part). Comparison between translations produced via correct (a) and incorrect (b) reorderings.

Figure 3: Sequences of bilingual tokens with source words substituted with their and their parents’ POS tags: correct (a) and incorrect (b) reorderings.

2.3 BiLM with dependency information

Dependency grammar is commonly used in NLP to formalize role-based relations between words. The intuitive notion of syntactic modification is captured by the primitive binary relation of dependence. Dependency relations do not change with the linear order of words (Figure 2) and therefore can provide a characterization of a word’s syntactic class that is invariant under reordering.

If we incorporate dependency relations into the representation of bilingual tokens, the incorrect reordering in Figure 2.b will produce a highly unlikely sequence. For example, we can substitute each source word with its POS tag and its parent’s POS tag (Figure 3). Again, we computed 4-gram log-probabilities for the corresponding sequences: the correct reordering results in a substantially higher probability of −10.58 than the incorrect one (−13.48). We may consider situations where more fine-grained distinctions are required. In the next section, we explore different representations based on source dependency trees.

3 Dependency-based BiLM

In this section, we introduce our model which combines the BiLM from Niehues et al. (2011) with source dependency information. We further give details on how the proposed models are trained and integrated into a phrase-based decoder.
3.1 The general framework

In the previous section we outlined our framework as composed of two steps: First, a parallel sentence is tokenized according to the BiLM model (Niehues et al., 2011). Next, words in the bilingual tokens are substituted with their contextual properties. It is thus convenient to use the following generalized definition for a token sequence \( t_1 \ldots t_n \) in our framework:

\[
t_i = (\text{ContE}(e_i), \{\text{ContF}(f) | f \in A(e_i)\}),
\]

where \( e_i \) is the \( i \)-th target word, \( A : E \rightarrow \mathcal{P}(F) \) is an alignment function, \( F \) and \( E \) are source and target sentences, and \( \text{ContE} \) and \( \text{ContF} \) are target and source contextual functions, respectively. A contextual function returns a word’s contextual property, based on its sentential context (source or target). See Figure 4 for an example of a sequence of BiLM tokens with a \( \text{ContF} \) defined as returning the POS tag of the source word combined with the POS tags of its parent, grandparent and siblings, and \( \text{ContE} \) defined as an identity function (see Section 3.2 for a detailed explanation of the functions and notation).

In this work we focus on source contextual functions (\( \text{ContF} \)). We also exploit some very simple target contextual functions, but do not go into an in-depth exploration.

3.2 Dependency-based contextual functions

In NLP approaches exploiting dependency structure, two kinds of relations are of special importance: the parent-child relation and the sibling relation. Shen et al. (2008) work with two well-formed dependency structures, both of which are defined in such a way that there is one common parent and a set of siblings. Li et al. (2012) characterize rules in hierarchical SMT by labeling them with the POS tags of the parents of the words inside the rule. Lerner and Petrov (2013) model reordering as a sequence of classification steps based on a dependency parse of a sentence. Their model first decides how a word is reordered with respect to its parent and then how it is reordered with respect to its siblings.

Based on these previous approaches, we propose to characterize contextual syntactic roles of a word in terms of POS tags of the words themselves and their relatives in a dependency tree. It is straightforward to incorporate parent information since each node has a unique parent. As for siblings information, we incorporate POS tags of the closest sibling to the left and the closest to the right. We do not include all of the siblings to avoid overfitting. In addition to these basic syntactic relations, we consider the grandparent relation.

The following list is a summary of the source contextual functions that we use. We describe a function with respect to the kind of contextual property of a word it returns: (i) the word itself (\( \text{Lex} \)); (ii) POS label of the word (\( \text{Pos} \)); (iii) POS label of the word’s parent; (iv) POS of the word’s closest sibling to the left, concatenated with the POS tag of the closest sibling to the right; (v) the POS label of the word’s grandparent. We use target-side contextual functions returning: (i) an empty string, (ii) POS of the word, (iii) the word itself.

**Notation.** We do not use the above functions separately to define individual BiLM models, but use combinations of these functions. We use the following notation for function combinations: “•” horizontally connects source (on the left) and target (on the right) contextual functions for a given model. For example, \( \text{Lex} \bullet \text{Lex} \) refers to the original (lexicalized) BiLM. We use arrows (\( \rightarrow \)) to designate parental information (the arrow goes from parent to child). \( \text{Pos} \rightarrow \text{Pos} \) refers to a combination of a function returning the POS of a word and the POS of its parent (as in Figure 3). \( \text{Pos} \rightarrow \text{Pos} \rightarrow \text{Pos} \) is a combination of the previous with the function returning the grandparent’s POS. Finally, we use +sibl to indicate the use of the sibling function described above: For example, \( \text{Pos} \rightarrow \text{Pos} + \text{sibl} \) is a source function that returns the word’s POS, its parent’s POS and the POS labels of the closest siblings to left and right.\(^6\) \( \text{Pos} + \text{sibl} \rightarrow \text{Pos} \) is a source function returning the word’s own POS, the POS of a word’s parent, and the POS tags of the parent’s siblings (left- and right-adjacent).

Figure 4 represents the sentence from Figure 2 during decoding in a system with an integrated \( \text{Pos} \rightarrow \text{Pos} \rightarrow \text{Pos} + \text{sibl} \bullet \text{Lex} \) feature. It shows the sequence of produced bilingual tokens and corresponding labels in the introduced notation.

3.3 Training

Training of dependency-based BiLMs consists of a sequence of extraction steps: After having produced word-alignments for a bitext (Section 4),

\(^6\) In case there is no sibling on one of the sides, \( \epsilon \) (empty word) is returned.
sentences are segmented according to Equation 3. We produce a dependency parse of a source sentence and a POS-tag labeling of a target sentence. For Chinese, we use the Stanford dependency parser (Chang et al., 2009). For Arabic a dependency parser is not available for public use, so we produce a constituency parse with the Stanford parser (Green and Manning, 2010) and extract dependencies based on the rules in Collins (1999). For English POS-tagging, we use the Stanford POS-tagger (Toutanova et al., 2003). After having produced a labeled sequence of tokens, we learn a 5-gram model using SRILM (Stolcke et al., 2011). Kneyser-Ney smoothing is used for all model variations except for $\text{Pos} \rightarrow \text{Pos}$ where Witten-Bell smoothing is used due to zero count-of-counts.

### 3.4 Decoder integration

Dependency-based BiLMs are integrated into our phrase-based SMT decoder as follows: Before translating a sentence, we produce its dependency parse. Phrase-internal word-alignments, needed to segment the translation hypothesis into tokens, are stored in the phrase table, based on the most frequent internal alignment observed during training. Likewise, we store the most likely target-side POS-labeling for each phrase pair.

The decoding algorithm is augmented with one additional feature function and one additional, corresponding feature weight. At each step of the derivation, as a new phrase pair is added to the partial translation hypothesis, this function segments the new phrase into bilingual tokens (given the internal alignment information) and substitutes the words in the phrase pair with syntactic labels (given the source parse and the target POS labeling associated with the phrase). The new syntactified bilingual tokens are added to the stack of preceding $n-1$ tokens, and the feature function computes the weighted updated model probability. During decoding, the probabilities of the BiLMs are computed in a stream-based fashion, with bilingual tokens as string tokens, and not in a class-based fashion, with syntactic source-side representations emitting the corresponding target words (Bisazza and Monz, 2014).

### 4 Experiments

#### 4.1 Setup

We conduct translation experiments with a baseline PBSMT system with additionally one of the dependency-based BiLM feature functions specified in Section 3. We compare the translation performance to a baseline PBSMT system and to a baseline augmented with the original BiLMs from (Niehues et al., 2011).

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 contains 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 Kneyser-Ney smoothing and interpolation. The BiLMs were trained as described in Section 3.3. Information about the parallel data used for training the Arabic-English$^7$ and Chinese-English systems$^8$ is

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

Table 1: Training data for Arabic-English and Chinese-English experiments.
### Table 2: BLEU and TER scores for Arabic-English experiments. Statistically significant improvements over the baseline (a) are marked ▲ at the \( p < .01 \) level and △ at the \( p < .05 \) level. Additionally, · ▲ and · △ indicate significant improvements with respect to BiLM \(Lex \bullet Lex\) (b). Since TER is an error rate, lower scores are better.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>MT08 BLEU</th>
<th>MT08 TER</th>
<th>MT09 BLEU</th>
<th>MT09 TER</th>
<th>MT08+MT09 BLEU</th>
<th>MT08+MT09 TER</th>
</tr>
</thead>
<tbody>
<tr>
<td>a PBSMT baseline</td>
<td>45.12</td>
<td>47.94</td>
<td>48.16</td>
<td>44.30</td>
<td>46.57</td>
<td>46.21</td>
</tr>
<tr>
<td>b Lex•Lex</td>
<td>45.27</td>
<td>47.79</td>
<td>48.85*</td>
<td>43.96*</td>
<td>46.98*</td>
<td>45.96*</td>
</tr>
<tr>
<td>Lex•Pos</td>
<td>44.80</td>
<td>47.84</td>
<td>48.22</td>
<td>44.14*</td>
<td>46.44</td>
<td>46.07</td>
</tr>
<tr>
<td>c Pos•Pos</td>
<td>45.66*</td>
<td>47.17*</td>
<td>49.00*</td>
<td>43.45*</td>
<td>47.25*</td>
<td>45.40*</td>
</tr>
<tr>
<td>d Pos•Lex</td>
<td>45.46*</td>
<td>47.45**</td>
<td>48.69*</td>
<td>43.64**</td>
<td>47.00*</td>
<td>45.64*</td>
</tr>
<tr>
<td>e Pos•Lex•Pos</td>
<td>45.68*</td>
<td>47.42**</td>
<td>49.09*</td>
<td>43.59*</td>
<td>47.30*</td>
<td>45.60*</td>
</tr>
<tr>
<td>f Lex•Lex + Pos•Pos</td>
<td>45.63**</td>
<td>47.48**</td>
<td>49.30**</td>
<td>43.60**</td>
<td>47.38**</td>
<td>45.63**</td>
</tr>
</tbody>
</table>

### Table 3: Different combinations of a target contextual function with the Pos—Pos source contextual function for Arabic-English. See Table 2 for the notation regarding statistical significance.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>MT08 BLEU</th>
<th>MT08 TER</th>
<th>MT09 BLEU</th>
<th>MT09 TER</th>
<th>MT08+MT09 BLEU</th>
<th>MT08+MT09 TER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pos—Pos• • •</td>
<td>45.66*</td>
<td>47.44*</td>
<td>48.78*</td>
<td>43.94*</td>
<td>47.15*</td>
<td>45.77*</td>
</tr>
<tr>
<td>Pos—Pos•Pos</td>
<td>45.66*</td>
<td>47.17**</td>
<td>49.06*</td>
<td>43.45*</td>
<td>47.25*</td>
<td>45.40**</td>
</tr>
<tr>
<td>Pos—Pos•Lex</td>
<td>45.48*</td>
<td>47.34**</td>
<td>48.90*</td>
<td>43.87*</td>
<td>47.12*</td>
<td>45.69**</td>
</tr>
</tbody>
</table>

The feature weights were tuned by using pairwise ranking optimization (Hopkins and May, 2011) on the MT04 benchmark (for both language pairs). During tuning, 14 PRO parameter estimation runs are performed in parallel on different samples of the n-best list after each decoder iteration. The weights of the individual PRO runs are then averaged and passed on to the next decoding iteration. Performing weight estimation independently for a number of samples corrects for some of the instability that can be caused by individual samples. For testing, we used MT08 and MT09 for Arabic, and MT06 and MT08 for Chinese. We use approximate randomization (Noreen, 1989; Riezler and Maxwell, 2005) to test for statistically significant differences.

In the next two subsections we discuss the general results for Arabic and Chinese, where we use case-insensitive BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) as evaluation metrics. This is followed by a preliminary analysis of observed reordering decisions where we compare 4-gram precision results and conduct experiments with an increased distortion limit.

### 4.2 Arabic-English translation experiments

We are interested in how a translation system with an integrated dependency-based BiLM feature performs as compared to the standard PBSMT baseline and, more importantly, to the original BiLM model. We consider two variants of BiLM discussed by Niehues et al. (2011): the standard one, \(Lex \bullet Lex\), and the simplest syntactic one, \(Pos \bullet Pos\). Results for the experiments can be found in Table 2. In the discussion below we mostly focus on the experimental results for the large, combined test set MT08+MT09.

Table 2.a–b compares the performance of the baseline and original BiLM systems. \(Lex \bullet Lex\) yields strongly significant improvements over the baseline for BLEU and weakly significant improvements for TER. Therefore, for the rest of the experiments we are interested in obtaining further improvements over \(Lex \bullet Lex\).

\(Pos—Pos \bullet Pos\) (Table 2.c) demonstrates the effect of adding minimal dependency information to a BiLM. It results in strongly significant improvements over the baseline and weak improvements over \(Lex \bullet Lex\) in terms of BLEU. We additionally ran experiments with the different target functions (Table 3). \(• Pos\) shows the highest results, and \(• •\) the lowest ones: this implies that a rather expressive source syntactic representation alone still benefits from target-side syntactic information. Below, our dependency-based systems only use \(• Pos\).

Next, we tested the effect of adding more source...
Table 4: BLEU and TER scores for Chinese-English PBSMT baseline and BiLM pipelines. See Table 2 for the notation regarding statistical significance.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>MT06</th>
<th>MT08</th>
<th>MT06+MT08</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU</td>
<td>TER</td>
<td>BLEU</td>
</tr>
<tr>
<td>a PBSMT baseline</td>
<td>31.89</td>
<td>57.79</td>
<td>25.53</td>
</tr>
<tr>
<td>b Lex•Lex</td>
<td>32.84*</td>
<td>57.40*</td>
<td>25.91*</td>
</tr>
<tr>
<td></td>
<td>32.31*</td>
<td>57.89</td>
<td>25.66</td>
</tr>
<tr>
<td>c Pos—Pos•Pos</td>
<td>32.86*</td>
<td>57.05*</td>
<td>26.09*</td>
</tr>
<tr>
<td>d Pos—Pos•sib•Pos</td>
<td>32.27*</td>
<td>56.63*</td>
<td>25.75</td>
</tr>
<tr>
<td>e Pos—Pos•Pos•Pos</td>
<td>33.09*</td>
<td>57.54</td>
<td>26.35*</td>
</tr>
<tr>
<td>f Lex•Lex + Pos—Pos•Pos•Pos</td>
<td>33.43*</td>
<td>57.00*</td>
<td>26.50*</td>
</tr>
</tbody>
</table>

Table 5: Different combinations of a target contextual function with the Pos—Pos source contextual function for Chinese-English. See Table 2 for the notation regarding statistical significance.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>MT06</th>
<th>MT08</th>
<th>MT06+MT08</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU</td>
<td>TER</td>
<td>BLEU</td>
</tr>
<tr>
<td>Pos—Pos•c</td>
<td>32.41*</td>
<td>57.42*</td>
<td>25.84</td>
</tr>
<tr>
<td>Pos—Pos•Pos</td>
<td>32.86*</td>
<td>57.05*</td>
<td>26.09*</td>
</tr>
<tr>
<td>Pos—Pos•Lex</td>
<td>32.60*</td>
<td>57.03*</td>
<td>25.72</td>
</tr>
</tbody>
</table>

dependency information. Pos—Pos+sib•Pos (Table 2.d) only improves over the PBSMT baseline (but also shows weak improvements over Lex•Lex for TER). It significantly degrades the performance with respect to the Pos—Pos•Pos system (Table 2.c). Pos—Pos—Pos•Pos (Table 2.e) shows the best results overall for BLEU, although it must be pointed out that the difference with Pos—Pos•Pos is very small. With respect to TER, Pos—Pos•Pos outperforms the grandparent variant.

So far, we can conclude that source parent information helps improve translation performance. Increased specificity of a parent (parent specified by a grandparent) tends to further improve performance. Up to now, we have only used syntactic information and obtained considerable improvements over Pos•Pos, surpassing the improvement provided by Lex•Lex. Can we gain further improvements by also adding lexical information? To this end, we conduct experiments combining the best performing dependency-based BiLM (Pos—Pos•Pos•Pos) and the lexicalized BiLM (Lex•Lex). We hypothesize that the two models improve different aspects of translation: Lex•Lex is biased towards improving lexical choice and Pos—Pos—Pos•Pos towards improving reordering. Combining these two models, we may improve both aspects. The metric results for the combined set indeed support this hypothesis (Table 2.f).

4.3 Chinese-English translation experiments

The results of the Chinese-English experiments are shown in Table 4. In the discussion below we mostly focus on the experimental results for the large, combined test set MT06+MT08. We observe the same general pattern for the Pos—Pos source function (Table 4.c) as for Arabic-English: the system with the cPos target function has the highest scores (Table 5). All of the Pos—Pos configurations show statistically significant improvements over the PBSMT baseline. For TER, two of the three Pos—Pos variants significantly outperform Lex•Lex. The system with sibling information (Table 4.d) obtains quite low BLEU results, just as in the Arabic experiments. On the other hand, its TER results are the highest overall. The system with the Pos—Pos—Pos•Pos function (Table 4.e) achieves the best results among dependency-based BiLMs for BLEU. Finally, combining Pos—Pos—Pos•Pos and Lex•Lex results in the largest and significant improvements over all competing systems for BLEU.

4.4 Preliminary analysis of reordering in translation experiments

In general, the experimental results show that using source dependency information yields consistent improvements for translating from Arabic and Chinese into English. On the other hand, we have pointed out some discrepancies between the two metrics employed, suggesting that different system configurations may improve different aspects.
of translation. To this end, we conducted some additional evaluations to understand how reordering is affected by the proposed features.

We use 4-gram precision as a metric of how much of the reference set word order is preserved. Table 6 shows the corresponding results for both languages. Just as in the previous two sections, configurations with parental information produce the best results. For Arabic, all of the dependency configurations outperform Lex•Lex. But the system with two feature functions, one of which is Lex•Lex, still obtains the best results, which may suggest that the lexicalized BiLM also helps to differentiate between word orders. For Chinese, Pos→Pos•Pos and the system combining the latter and Lex•Lex also obtain the best results. However, other dependency-based configurations do not outperform Lex•Lex.

All the experiments so far were run with a distortion limit of 5. But both of the languages, especially Chinese, often require reorderings over a longer distance. We performed additional experiments with a distortion limit of 10 for the Lex•Lex and Pos→Pos•Pos systems (Tables 7 and 8). It is more difficult to translate with a higher distortion limit (Green et al., 2010) as the set of permutations grows larger thereby making it more difficult to differentiate between correct and incorrect continuations of the current hypothesis. It has also been noted that higher distortion limits are more likely to result in improvements for Chinese rather than Arabic to English translation (Chiang, 2007; Green et al., 2010).

We compared performance of fixed BiLM models at distortion lengths of 5 and 10. Arabic-English results did not reveal statistically significant differences between the two distortion limits for Pos→Pos•Pos. On the other hand, for Lex•Lex BLEU decreases when using a distortion limit of 10 compared to a limit of 5. This implies that the dependency BiLM is more robust in the more challenging reordering setting than the lexicalized BiLM. Chinese-English results for Pos→Pos•Pos do show significant improvements over the distortion limit of 5 (up to 0.49 BLEU higher than the best result in Table 4). This indicates that the dependency-based BiLM is better capable to take advantage of the increased distortion limit and discriminate between correct and incorrect reordering choices.

Comparing the results for Pos→Pos•Pos and Lex•Lex at a distortion limit of 10, we obtain strongly significant improvements for all metrics. For Chinese, a larger distortion limit helps for both configurations, but more so for our dependency BiLM, yielding an improvement of 0.98 BLEU.
over the original, lexicalized BiLM (Table 8).

5 Conclusions

In this paper, we have introduced a simple, yet effective way to include syntactic information into phrase-based SMT. Our method consists of enriching the representation of units of a bilingual language model (BiLM). We argued that the very limited contextual information used in the original bilingual models (Niehues et al., 2011) can capture reorderings only to a limited degree and proposed a method to incorporate information from a source dependency tree in bilingual units. In a series of translation experiments we performed a thorough comparison between various syntactically-enriched BiLMs and competing models. The results demonstrated that adding syntactic information from a source dependency tree to the representations of bilingual tokens in an n-gram model can yield statistically significant improvements over the competing systems.

A number of additional evaluations provided an indication for better modeling of reordering phenomena. The proposed dependency-based BiLMs resulted in an increase in 4-gram precision and provided further significant improvements over all considered metrics in experiments with an increased distortion limit.

In this paper, we have focused on rather elementary dependency relations, which we are planning to expand on in future work. Our current approach is still strictly tied to the number of target tokens. In particular, we are interested in exploring ways to better capture the notion of syntactic cohesion in translation (Fox, 2002; Cherry, 2008) within our framework.

Acknowledgments

We thank Arianna Bisazza and 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


