Anchor Graph:
Global Reordering Contexts for Statistical Machine Translation

Hendra Setiawan *
IBM Research
1101 Kitchawan Road
NY 10598, USA

Bowen Zhou
IBM Research
1101 Kitchawan Road
NY 10598, USA

Bing Xiang *
Thomson Reuters
3 Times Square
NY 10036, USA

Abstract

Reordering poses one of the greatest challenges in Statistical Machine Translation research as the key contextual information may well be beyond the confine of translation units. We present the "Anchor Graph" (AG) model where we use a graph structure to model global contextual information that is crucial for reordering. The key ingredient of our AG model is the edges that capture the relationship between the reordering around a set of selected translation units, which we refer to as *anchors*. As the edges link anchors that may span multiple translation units at decoding time, our AG model effectively encodes global contextual information that is previously absent. We integrate our proposed model into a state-of-the-art translation system and demonstrate the efficacy of our proposal in a large-scale Chinese-to-English translation task.

1 Introduction

Reordering remains one of the greatest challenges in Statistical Machine Translation (SMT) research as the key contextual information may span across multiple translation units.[1] Unfortunately, previous approaches fall short in capturing such cross-unit contextual information that could be critical in reordering. For example, state-of-the-art translation models, such as Hiero (Chiang, 2005) or Moses (Koehn et al., 2007), are good at capturing local reordering within the confine of a translation unit, but their formulation is approximately a simple unigram model over derivation (a sequence of the application of translation units) with some aid from target language models. Moving to a higher order formulation (say to a bigram model) is highly impractical for several reasons: 1) it has to deal with a severe *sparsity issue* as the size of the unigram model is already huge; and 2) it has to deal with a *spurious ambiguity issue* which allows multiple derivations of a sentence pair to have radically different model scores.

In this paper, we develop "*Anchor Graph*" (AG) where we use a graph structure to capture global contexts that are crucial for translation. To circumvent the sparsity issue, we design our model to rely only on contexts from a set of selected translation units, particularly those that appear frequently with important reordering patterns. We refer to the units in this special set as *anchors* where they act as vertices in the graph. To address the spurious ambiguity issue, we insist on computing the model score for every anchors in the derivation, including those that appear inside larger translation units, as such our AG model gives the same score to the derivations that share the same reordering pattern.

In AG model, the actual reordering is modeled by the edges, or more specifically, by the edges' labels where different reordering around the anchors would correspond to a different label. As detailed later, we consider two distinct set of labels, namely *dominance* and *precedence*, reflecting the two dominant views about reordering in literature, i.e. the first one that views reordering as a linear operation over a sequence and the second one that views reordering as a recursive operation over nodes in a tree structure The former is prevalent in phrase-based context, while the latter in hierarchical phrase-based and

* This work was done when the authors were with IBM.

[1] We define translation units as phrases in phrase-based SMT or as translation rules in syntax-based SMT.

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 501–512,
Seattle, Washington, USA, 18-21 October 2013.

syntax-based context. More concretely, the dominance looks at the anchors' relative positions in the translated sentence, while the precedence looks at the anchors' relative positions in a latent structure, induced via a novel synchronous grammar: *Anchor-centric, Lexicalized Synchronous Grammar*.

From these two sets of labels, we develop two probabilistic models, namely the *dominance* and the *orientation* models. As the edges of AG link pairs of anchors that may appear in multiple translation units, our AG models are able to capture high order contextual information that is previously absent. Furthermore, the parameters of these models are estimated in an unsupervised manner without linguistic supervision. More importantly, our experimental results demonstrate the efficacy of our proposed AG-based models, which we integrate into a state-of-the-art syntax-based translation system, in a large scale Chinese-to-English translation task. We would like to emphasize that although we use a syntax-based translation system in our experiments, in principle, our approach is applicable to other translation models as it is agnostic to the translation units.

2 Anchor Graph Model

Formally, an AG consists of $\{\mathcal{A}, \mathcal{L}\}$ where $\mathcal{A}$ is a set of vertices that correspond to anchors, while $\mathcal{L}$ is a set of labeled edges that link a pair of anchors. In principle, our AG model is part of a translation model that focuses on the reordering within the source sentence F and its translation E. Thus, we start by first introducing $\mathcal{A}$ into a translation model (either word-based, phrase-based or syntax-based model) followed by $\mathcal{L}$. Given an F, $\mathcal{A}$ is essentially a subset of non-overlapping (word or phrase) units that make up F. As the information related to $\mathcal{A}$ is not observed, we introduce $\mathcal{A}$ as a latent variable.

Let $P(E, \sim |F)$ be a translation model where $\sim$ corresponds to the alignments between units in F and E. [2] We introduce $\mathcal{A}$ into a translation model, as follow:

$$P(E, \sim |F) = \sum_{\forall \mathcal{A}'} P(E, \sim, \mathcal{A}'|F) \qquad (1)$$

$$P(E, \sim, \mathcal{A}'|F) = P(E, \sim |\mathcal{A}', F)P(\mathcal{A}') \qquad (2)$$

As there can be many possible subsets of F and summing over all possible $\mathcal{A}$ is intractable, we make the following approximation for $P(\mathcal{A}')$ such that we only need to consider one particular $\mathcal{A}^*$: $P(\mathcal{A}') = \delta(\mathcal{A}' = \mathcal{A}^*)$ which returns 1 only for $\mathcal{A}^*$, otherwise 0. The exact definition of the heuristic will be described in Section 7, but in short, we equate $\mathcal{A}^*$ with units that appear frequently with important reordering patterns in training data.

Given an $\mathcal{A}^*$, we then introduce the edges of AG $(\mathcal{L})$ into the equation as follow:

$$P(E, \sim |\mathcal{A}^*, F) = P(E, \sim, \mathcal{L}|\mathcal{A}^*, F) \qquad (3)$$

Note that $\mathcal{L}$ is also a latent variable but its values are derived deterministically from $(F, E, \sim)$ and $\mathcal{A}^*$, thus no extra summation is present in Eq. 3.

Then, we further simplify Eq. 3 by factorizing it with respect to each individual edges, as follow:

$$P(E, \sim, \mathcal{L}|\mathcal{A}^*, F) \approx \prod_{\substack{\forall a_m, a_n \in \mathcal{A}^* \\ m<n}} P(L_{m,n}|a_m, a_n) \qquad (4)$$

where $L_{m,n} \in \mathcal{L}$ corresponds to the label of an edge that links a_m and a_n.

In principle, $L_{m,n}$ can take any arbitrary value. For addressing the reordering challenge, it should ideally correspond to some aspect of the reordering around a_m and a_n, for example, how the reordering around a_m affects the reordering around a_n. As mentioned earlier, we choose to associate $L_{m,n}$ with the dominance and the precedence relations between a_m and a_n, where the former looks at the relative positions of the two anchors when they are projected into a latent tree structure, while the latter looks at their relative positions when they are projected into the target sentence. We illustrate the two in Fig. 1.

Furthermore, we assume that dominance and precedence are independent and develop one model for each, resulting in the dominance and the orientation models, which we describe in Section 3 and 4 respectively. To make the model more compact, we

[2] Alignment ($\sim$) represents an existing latent variable. Depending on the translation units, it can be defined at different level, i.e. word, phrase or hierarchical phrase. As during translation, we are interested in the anchors that appear inside larger translation units, we set $\sim$ at word level, which information can be induced for (hierarchical) phrase units by either keeping the word alignment from the training data inside the units or inferring it via lexical translation probability. We use the former.

introduce an additional parameter O that restricts the maximum order of AG as follows:

$$\approx \prod_{o=1}^{O} \prod_{i=0}^{|\mathcal{A}^*|+o-1} P_o(L_{i-o,i}|a_{i-o}, a_i) \qquad (5)$$

Thus, we only consider edges that link two anchors that are at most $O-1$ anchors apart. For $O = 1$, the AG model only considers relations between neighboring anchors. Following the standard practice in the n-gram language modeling, we append O number of pseudo anchors at the beginning and at the end of F, which represent the sentence delimiter markers. We do so in a monotone order.

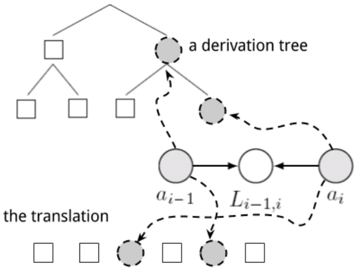

Figure 1: The illustration of the dominance and the precedence relations. The former looks at the anchors' projection on a derivation structure. The latter looks at the anchors' projection on the translated sentence.

3 Dominance Model

This section describes our dominance model where we equate $L_{m,n}$ in Eq. 4 with $dom(a_m, a_n)$ that expresses to the dominance relation between a_m and a_n in a latent tree structure. Due to reordering, anchors can only appear in specific nodes. We first describe a novel formalism of *Anchor-centric, Lexicalized Synchronous Grammar* (AL-SG), used to induce the tree structure and then discuss the probabilistic formulation of the model. Just to be clear, we introduce AL-SG mainly to facilitate the computation of $dom(a_m, a_n)$. The actual translation model at decoding time remains either phrase-based, hierarchical phrase-based or syntax-based model.

3.1 Anchor-centric, Lexicalized Synchronous Grammar

Given $(F, E, \sim)$ and $\mathcal{A}$, Anchor-centric, Lexicalized Synchronous Grammar (AL-SG) produces a tree structure where the nodes are decorated with anchors-related information. As the name alludes, the core of AL-SG is *anchor-centric constituents* (*ACC*), which corresponds to nodes, composed from merging anchors with by either their left, their right neighboring constituents or both.

More concretely, first of all, we consider a span on the source sentence F to be a *constituent* if it is consistent with the alignment ($\sim$). Second of all, we can construct a larger constituent by merging smaller constituents given that the larger constituent is also consistent with the alignment. These two constraints are similar to the heuristic applied to extract hierarchical phrases (Chiang, 2005).

Then, specific to AL-SG, we consider an anchor a to lexicalize a constituent c, if: a) we can compose c from at most three smaller constituents: c_L, a and c_R where a is the anchor while c_L,c_R are the (possibly empty) constituents immediately to the left and to the right of a; and b) we can create smaller anchors-centric constituents from concatenating a with c_L and a with c_R. If a can lexicalize c, then the node associated with c would be marked with a. In computing $dom(a_m, a_n)$, we look at the constituents that cover both anchors and check whether the anchors can lexicalized any of such constituents.

Now, we will describe AL-SG in a formal way. For simplicity, we use a simple grammar, called Inversion Transduction Grammar (ITG) (Wu, 1997), although in practice, we handle a more powerful synchronous grammar. Hence, we proceed to describe Anchor-centric, Lexicalized ITG (AL-ITG).

An AL-ITG is a quadruple $\{\Sigma, \mathcal{A}, \mathcal{V}, \mathcal{R}\}$ where:

- $\Sigma = \{(f/e)\}$ is a set of terminal symbols, which represents all possible units defined over $(F, E, \sim)$ where each pair corresponds to a link in $\sim$. We define $\sim$ at the most fine-grained level (i.e. word-level), as we insist on computing model score for each anchors even if they appear inside larger units.
- $\mathcal{A} \in \Sigma$ is a set of anchors, which is a subset of the terminal symbols.
- $\mathcal{V} = \{\{P, X, Y\} \times \{\mathcal{A}, \emptyset\}\}$ is a set of (possibly lexicalized) nonterminal symbols. P represents the terminal symbols (Σ); while X and Y correspond to the spans that are created from merging two adjacent constituents. On the tar-

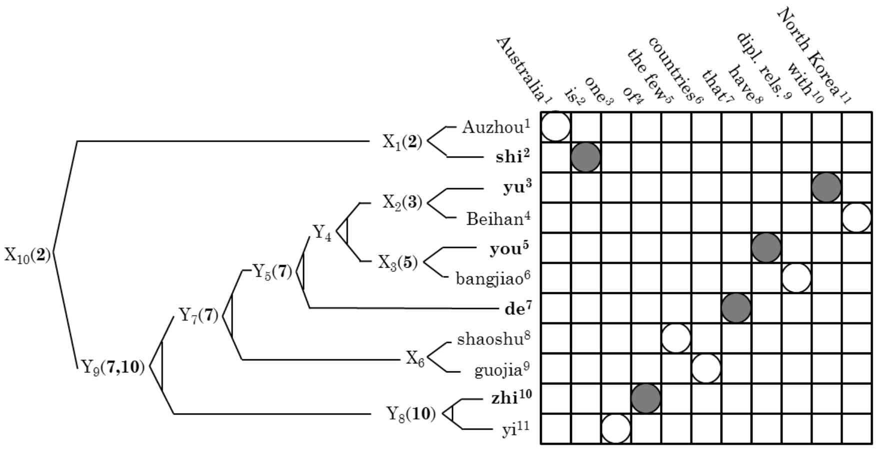

Figure 2: An illustration of an aligned Chinese-English sentence pair with one possible AL-ITG derivation obtained by applying the grammar in a left-to-right fashion. Circles represent alignment points. Black circle represents the anchor; boxes represent the anchor's neighbors. In the derivation tree, the anchors are represented by their position and in bold. For succinctness, we omit the preterminal rules in the tree.

get side, for X, the order of the two children follows the source order, while for Y, the order follows the inverse. Nonterminal symbols can be lexicalized with zero or more than one anchor. We represent a lexicalized constituent as a nonterminal symbol followed by a bracket which contains the lexicalizing anchors, e.g. $P(H)$ where H is the anchors lexicalizing P.

- $\mathcal{R}$ is a set of production rules which can be classified into the following categories:
  - Preterminal rules. We propagate the symbol if it corresponds to an anchor.

$$P(H = f/e) \rightarrow f/e, \text{if } f/e \in \mathcal{A}^*$$
$$P(H = \emptyset) \rightarrow f/e, \text{otherwise}$$

  - Monotone production rules, which reorder the children in monotone order, denoted by square brackets ("[","]").

$$\begin{aligned}
X(H_1 \cup H_2) \rightarrow &\quad [P(H_1)P(H_2)] \\
X(H_1 \cup H_2) \rightarrow &\quad [X(H_1)P(H_2)] \\
X(H_1 \cup H_2) \rightarrow &\quad [X(H_1)X(H_2)] \\
X(H_1) \rightarrow &\quad [X(H_1)Y(H_2)] \\
X(H_2) \rightarrow &\quad [Y(H_1)P(H_2)] \\
X(H_2) \rightarrow &\quad [Y(H_1)X(H_2)] \\
X(\emptyset) \rightarrow &\quad [Y(H_1)Y(H_2)]
\end{aligned}$$

  - Inverse production rules, which reorder the children in the inverse order, denoted by angle brackets ("⟨","⟩").

$$\begin{aligned}
Y(H_1 \cup H_2) \rightarrow &\quad \langle P(H_1)P(H_2)\rangle \\
Y(H_1 \cup H_2) \rightarrow &\quad \langle Y(H_1)P(H_2)\rangle \\
Y(H_1 \cup H_2) \rightarrow &\quad \langle Y(H_1)Y(H_2)\rangle \\
Y(H_1) \rightarrow &\quad \langle Y(H_1)X(H_2)\rangle \\
Y(H_2) \rightarrow &\quad \langle X(H_1)P(H_2)\rangle \\
Y(H_2) \rightarrow &\quad \langle X(H_1)Y(H_2)\rangle \\
Y(\emptyset) \rightarrow &\quad \langle X(H_1)X(H_2)\rangle
\end{aligned}$$

Like ITG, AL-ITG only permits two kind of reordering operations, namely monotone and inverse. To accommodate the lexicalization, we first assign a unique nonterminal symbol for each, i.e. X for monotone reordering and Y for inverse reordering. Then, we lexicalize Xs and Ys with anchors as long as they satisfy the constraint that the child shares the same label as the parent. This constraint guarantees that the constituents are valid ACCs. It also enables the anchors to lexicalize long constituents, although the terminal symbols are defined at word-level.

Fig. 2 illustrates an example Chinese-to-English translation with a AL-ITG derivation when the grammar is applied in a left-to-right fashion. Admittedly, AL-ITG (or more generally AL-SG) is susceptible to spurious ambiguity as it produces multiple derivation trees for a given $(F, E, \sim)$. Fortunately, the value of $dom(a_m, a_n)$ is identical for all derivations, since the computation of $dom(a_m, a_n)$ relies only on whether a_m and a_n can lexicalize *at least one* constituent that covers both anchors. Hence, we only need to look at one derivation to compute $dom(a_m, a_n)$. Generalizing AL-ITG to a more powerful formalism is trivial; we just need to forbid the propagation for non-binarizeable production rules.

3.2 Probabilistic Model

We read-off the dominance relations $dom(a_m, a_n)$ from D obtained from the application of AL-SG to $(F, E, \sim)$. As lexicalization is a bottom-up process, for reading-off $dom(a_m, a_n)$, it is sufficient to look at the lowest common ancestor (LCA) of both anchors; if the anchors cannot lexicalize the LCA, they won't be able to lexicalize the constituents larger than LCA. To be more concrete, let's consider the D in Fig. 2. In that D, the LCA of $a_m = \text{yu}^3/\text{with}^{10}$ and $a_n = \text{de}^7/\text{that}^7$ is $Y_5(\mathbf{7})$. Then, we check the anchors that can lexicalize the LCA. Let $V(H)$ be the LCA, then $dom(a_m, a_n) \in$

$$\begin{array}{rl} (\mathit{LH}) & \text{, if } a_m \in H \wedge a_n \notin H \\ (\mathit{RH}) & \text{, if } a_m \notin H \wedge a_n \in H \\ (\mathit{BL}) & \text{, if } a_m \in H \wedge a_n \in H \\ (\mathit{BD}) & \text{, if } a_m \notin H \wedge a_n \notin H \end{array}$$

The value refers to cases where a_m and a_n can lexicalize $V(H)$ and it is useful to model spans that share a simple, uniform reordering, i.e. all-monotone or all-inverse, while the value refers to the cases where a_m and a_n cannot lexicalize $V(H)$ and it is useful to model spans that involve in a complex reordering. Meanwhile, the and refer to cases where only one anchor can lexicalize $V(H)$, i.e. a_m and a_n respectively. These values are useful for modeling cases where the surroundings of the two anchors exhibit different kind of reordering pattern.

With such definition, the edge labels $\mathcal{L}$ in Fig. 2 are indicated in Table 1. Note that in Table 1, we don't specify the relations involving pseudo anchors, although they are crucial.

The final probabilistic formulation of the dominance model is as follows:

$$\approx \prod_{o=1}^{O} \prod_{i=0}^{|\mathcal{A}|+o-1} P_{dom_o}(dom(a_{i-o}, a_i)|a_{i-o}, a_i) \quad (6)$$

As shown, we allocate a separate model P_{dom_o} for each separate order (o) where each P_{dom_o} will contribute as one additional feature in the log-linear model of the translation model. In allocating a separate model for each o, we conjecture that different pair of anchors contributes differently depending on how far the two anchors are.

m n	1	2	3	4	5
1 = (shi^2/is^2)	-	-	-	-	-
2 = (yu^3/with^{10})	*LH*	-	-	-	-
3 = (you^5/have^8)	*LH*	*BD*	-	-	-
4 = (de^7/that^7)	*LH*	*RH*	*RH*	-	-
5 = (zhi^{10}/of^4)	*LH*	*RH*	*RH*	*BL*	-

Table 1: The dominance relations between pairs of anchors according to the derivation in Fig. 2.

4 Orientation Model

In this section, we introduce the orientation model (*ori*) where we equate $L_{m,n}$ with the precedence relations between a pair of anchors. Instead of directly modeling the precedence between the two anchors, we approximate it by modeling the precedence of each anchor with its neighboring constituents. Formally, we approximate $P(L_{m,n}|a_m, a_n)$ as

$$\begin{aligned} &P_{ori_R}(ori(a_m, M_R(a_m))|a_m) \times \\ &\quad P_{ori_L}(ori(a_n, M_L(a_n))|a_n) \end{aligned} \quad (7)$$

where $M_R(a_m)$ is the largest constituent to the right of the first anchor a_m, $M_L(a_n)$ the largest constituent to the left of the second anchor a_n, and $ori()$ a function that maps the anchor and the neighboring constituent to a particular orientation.

Plugging Eq. 7 into Eq. 5 results in the following approximation of $P(\Theta|\mathcal{A})$:

$$\begin{aligned} C. \prod_{i=0}^{|\mathcal{A}|-1} \quad &\{P_{ori_L}(ori(a_i, M_L(a_i))|a_i) \times \\ &P_{ori_R}(ori(a_i, M_R(a_i))|a_i)\}^O \end{aligned} \quad (8)$$

where C is a constant term related to the pseudo anchors and O is the maximum order of the AG. In practice, we can safely ignore both C and O as they are constant for a given AG. As shown, the orientation model is simplified into a model that looks at the reordering of the anchors' neighboring constituents.

The exact definition of M_L and M_R will be discussed in Section 5. Their orientation, i.e.

$ori_L(C_L, a)$ and $ori_R(C_R, a)$ respectively, may take one of the following four values: (*MA*), (*RA*), (*MG*) and (*RG*). The first clause (monotone, reverse) indicates whether the target order follows the source order; the second (adjacent, gap) indicates whether the anchor and its neighboring constituent are adjacent or separated by an intervening when projected.

5 Parameter Estimation

For each $(F, E, \sim)$, the training starts with the identification of the regions in the source sentences as anchors ($\mathcal{A}$). For our Chinese-English experiments, we use a simple heuristic that equates anchors ($\mathcal{A}^*$) with constituents whose corresponding word class belongs to function words-related classes, bearing a close resemblance to (Setiawan et al., 2007). In total, we consider 21 part-of-speech tags; some of which are as follows: VC (copula), DEG, DEG, DER, DEV (*de*-related), PU (punctuation), AD (adjectives) and P (prepositions).

5.1 Extracting Events from $(F, E, \sim)$

The parameter estimation first involves extracting two statistics from $(F, E, \sim)$, namely $dom(a_m, a_n)$ for the dominance model as well as $ori(a, M_L(a))$ and $ori(a, M_R(a))$ for the orientation model. Instead of developing a separate algorithm for each, we describe a unified way to extract these statistics via the largest neighboring constituents of the anchors, i.e. $M_L(a)$ and $M_R(a)$. This approach enables the dominance model to share the same residual state information as the orientation model.[3]

Let a_m be an anchor and $M_R(a_m)$ be its largest neighboring constituent to the right. Let a_n be an anchor to the left of a_m and $M_L(a_n)$ be a_n's largest neighboring constituent to the left. According to AL-SG, we say that a_m dominates a_n if $ori(a_m, M_R(a_m)) \in \{MA, RA\}$ and $a_n \in M_R(a_m)$. By the same token, we say that a_n dominates a_m if $ori(a_n, M_L(a_n)) \in \{MA, RA\}$ and $a_m \in M_L(a_n)$. The constraints on the orientation reflect the fact that in AL-SG, anchors can only be propagated through monotone or inverse production rules, which correspond to the MA and RA respectively. The fact that we are looking at the largest neighboring constituents guarantees that if the other anchor is outside that constituent, then that other anchor is never dominated.

More formally, given an aligned sentence pair $\Theta = (F, E, \sim)$, let $\Delta(\Theta)$ be all possible constituents that can be extracted from Θ:[4]

$$\{(f_{j_1}^{j_2}/e_{i_1}^{i_2}) : \forall (j,i) \in \sim : ((j_1 \leq j \leq j_2) \wedge (i_i \leq i \leq i_2)) \\ \vee (\neg(j_1 \leq j \leq j_2) \wedge \neg(i_i \leq i \leq i_2))$$

Then, let the anchors $\mathcal{A}$ be a subset of $\Delta(\Theta)$. Given $\mathcal{A} \subset \Delta(\Theta)$, let $a = (f_{j_1}^{j_2}/e_{i_1}^{i_2}) \in \mathcal{A}$ be a particular anchor. And, let $\mathcal{C}_L(a) \subset \Delta(\Theta)$ be a's left neighbors and let $\mathcal{C}_R(a) \subset \Delta(\Theta)$ be a's right neighbors, iff:

$$\forall C_L = (f_{j_3}^{j_4}/e_{i_3}^{i_4}) \in \mathcal{C}_L(a) : j_4 + 1 = j_1$$
$$\forall C_R = (f_{j_5}^{j_6}/e_{i_5}^{i_6}) \in \mathcal{C}_R(a) : j_2 + 1 = j_5$$

Then, let $M_L(a) \in \mathcal{C}_L(a)$ and $M_R(a) \in \mathcal{C}_R(a)$ be the largest left and right neighbors according to:

$$M_L(a) = \underset{(f_{j_3}^{j_4}/e_{i_3}^{i_4}) \in \mathcal{C}_L(a)}{\arg\max} (j_4 - j_3)$$
$$M_R(a) = \underset{(f_{j_5}^{j_6}/e_{i_5}^{i_6}) \in \mathcal{C}_R(a)}{\arg\max} (j_6 - j_5)$$

Let $M_L = (f_{j_3}^{j_4}/e_{i_3}^{i_4})$ and $M_R = (f_{j_5}^{j_6}/e_{i_5}^{i_6})$. We then proceed to extract $ori_L(a, M_L(a))$ and $ori_R(a, M_R(a))$ respectively as follows:

- *MA*, if $(i_4 + 1) = i_1$ for ori_L or if $(i_2 + 1) = i_5$ for ori_R
- *RA*, if $(i_2 + 1) = i_3$ for ori_L or if $(i_6 + 1) = i_1$ for ori_R
- *MG*, if $(i_4 + 1) < i_1$ for ori_L or if $(i_2 + 1) < i_5$ for ori_R
- *RG*, if $(i_2 + 1) < i_3$ for ori_L or if $(i_6 + 1) < i_1$ for ori_R.

Then, we proceed to extract $dom(a_m, a_n)$. Given two anchors a_m, a_n where $m < n$, we define the

[3] The analogy in an n-gram language model is the first $n - 1$ words of the hypothesis that have incomplete history.

[4] We represent a constituent as a source and target phrase pair $(f_{j_1}^{j_2}/e_{i_1}^{i_2})$ where the subscript and the superscript indicate the starting and the ending indices as such $f_{j_1}^{j_2}$ denotes a source phrase that spans from j_1 to j_2.

dominance relation between a_m and a_n via $M_R(a_m)$ and $M_L(a_n)$. Let $a_m = (f_{j_1}^{j_2}/e_{i_1}^{i_2})$, $M_R(a_m) = (f_{j_3}^{j_4}/e_{i_3}^{i_4})$, $a_n = (f_{j_5}^{j_6}/e_{i_5}^{i_6})$ and $M_L(a_n) = (f_{j_7}^{j_8}/e_{i_7}^{i_8})$. Then, $ldom(a_m, a_n)$ is true only if $(j_4 \geq j_6)$ and $ori_R(a_m, M_R(a_m)) \in \{MA, RA\}$. Similarly, $rdom(a_m, a_n)$ is true only if $(j_7 \leq j_1)$ and $ori_L(a_n, M_L(a_n)) \in \{MA, RA\}$.

Hence, $dom(a_m, a_n)$ is as follows:

- *LH*, if $ldom(a_m, a_n) \wedge \neg rdom(a_m, a_n)$
- *RH*, if $\neg ldom(a_m, a_n) \wedge rdom(a_m, a_n)$
- *BL*, if $ldom(a_m, a_n) \wedge rdom(a_m, a_n)$
- *BD*, if $\neg ldom(a_m, a_n) \wedge \neg rdom(a_m, a_n)$

5.2 Parameterization and Training

After extracting events, we are now ready to train the models. To estimate them, we train a discriminative classifier for each model and use the normalized posteriors at decoding time as additional feature scores in SMT's log-linear framework.

At a high level, we use a rich set of *binary* features ranging from lexical to part-of-speech (POS) and to syntactic features. Additionally, we augment the feature set with compound features, e.g. a conjunction of the source word of the left anchor and the source word of the right anchor. Although they increase the number of features significantly, we found that they are empirically beneficial.

Suppose $a = (f_{j_1}^{j_2}/e_{i_1}^{i_2})$, $M_L(a) = (f_{j_3}^{j_4}/e_{i_3}^{i_4})$ and $M_R(a) = (f_{j_5}^{j_6}/e_{i_5}^{i_6})$, then based on the context's location, the elementary features employed in our classifiers can be categorized into:

- *anchor-related*: (the actual word of $f_{j_1}^{j_2}$), (part-of-speech (POS) tag of), ('s parent in the parse tree), ($e_{i_1}^{i_2}$'s actual target word).
- *surrounding*: (the previous word / $f_{j_1-1}^{j_1-1}$), (the next word / $f_{j_2+1}^{j_2+1}$), ('s POS tag), ('s POS tag), ('s parent), ('s parent).
- *non-local*: (the previous anchor's source word) , (the next anchor's source word), ('s POS tag), ('s POS tag).

There is a separate set of elementary features for a_m and a_n and we come up with manual combination to construct compound features.

In training the models, we manually come up with around 30-50 types of features, which consists of a combination of elementary and compound features. Due to space constraints, we will describe the actual features that we use and the classification performance of our models elsewhere. In total, we generate around one hundred millions binary features from our training data that contains six million sentence pairs. To reduce the number of features, we employ the L1-regularization in training to enforce sparse solutions, using the off-the-shelf LIBLINEAR toolkit (Fan et al., 2008). After training, the number of features in our classifiers decreases to below 1 million features for each classifier.

6 Decoding

As mentioned earlier, we wish to avoid the spurious ambiguity issue where different derivations have radically different scores although they lead to the same reordering. This section describes our decoding algorithm that avoids spurious ambiguity issue by incrementally constructing M_Ls and M_Rs thus allowing the computation of the models over partial hypotheses.

In our experiments, we integrate our dominance model as well as our orientation model into a syntax-based SMT system that uses SCFG formalism. Integrating the models into syntax-based SMT systems is non-trivial, especially since the anchors often reside within translation rules and the model doesn't always decompose naturally with the hypothesis structure. To facilitate that, we need to first induce the necessary alignment for all translation units in the hypothesis.

To describe the algorithm, let us consider a cheating exercise where we have to translate the Chinese sentence in Fig. 2 with the following set of hierarchical phrases:

$$\begin{aligned}
&X_a \rightarrow \langle \textit{Aozhou}^1 \textit{shi}^2\, X_1, \text{Australia}^1 \text{ is}^2 X_1 \rangle \\
&X_b \rightarrow \langle \textit{yu}^3\, \textit{Beihan}^4\, X_1, X_1 \text{with}^3 \text{ North}^4 \text{ Korea} \rangle \\
&X_c \rightarrow \langle \textit{you}^5 \textit{bangjiao}^6, \text{have}^5 \text{dipl.}^6 \text{ rels.} \rangle \\
&X_d \rightarrow \langle X_1\, \textbf{\textit{de}}^7 \textit{shaoshu}^8\, \textit{guojia}^9\, \textit{zhi}^{10}\, \textit{yi}^{11}, \\
&\qquad\quad \text{one}^{11} \text{of}^{10} \text{the few}^8 \text{ countries}^9\, \textbf{that}^7 X_1 \rangle
\end{aligned}$$

As a case in point, let us consider $D = X_a \prec X_b \prec X_d \prec X_c$, which will lead to the correct English

		Target string (w/ source index)	Symbol(s) read	Op.	Stack(s)
(1)	X_c	have[5] dipl.[6] rels.	[5][6]	S,S,R	X_c:[5-6]
(2)	X_d	one[11] of[10] few[8] countries[9] **that**[7] X_c	[11][10]	S,S,R	[10-11]
(3)			[8][9]	S,S,R,R	[8-11]
(4)			[7]	S	[8-11]**[7]**
(5)			X_c:[5,6]	S	X_d:[8-11]**[7]**[5,6]
(6)	X_b	X_d with[3] North[4] Korea	X_d:[8-11]**[7]**[5,6]	S	[8-11]**[7]**[5,6]
(7)			[3][4]	S,S,R,R	X_b:[8-11]**[7]**[3-6]
(8)	X_a	Australia[1] is[2] X_b	[1][2]	S,S,R	[1-2]
(9)			X_b:[8-11]**[7]**[3,6]	S,A	X_a:[1-2][8-11]**[7]**[3,6]

Table 2: The application of the shift-reduce parsing algorithm, which corresponds to the following derivation $D = X_a \prec X_b \prec X_d \prec X_c$. Anchor is in **bold**. In column Op., S, R and A refer to shift, reduce and accept operation respectively.

translation as in Fig. 2. Note that the translation rules contain internal word alignment, which we assume to have been previously inferred.

The algorithm bears a close resemblance to the shift-reduce algorithm found in phrase-based decoding (Galley and Manning, 2008; Feng et al., 2010; Cherry et al., 2012). A stack is used to accumulate (partial) information about a, M_L and M_R for each $a \in \mathcal{A}$ in the derivation. This algorithm takes an input stream and applies either the *shift* or the *reduce* operations starting from the beginning until the end of the stream. The *shift* operation advances the input stream by one symbol and push the symbol into the stack; while the *reduce* operation applies some rule to the top-most elements of the stack. The algorithm terminates at the end of the input stream where the resulting stack will be propagated to the parent for the later stage of decoding. In our case, the input stream is the target string of the rule and the symbol is the corresponding source index of the elements of the target string. The reduction rule looks at two indices and merge them if they are adjacent (i.e. has no intervening phrase). We forbid the application of the reduction rule to anchors. Table 2 shows the execution trace of the algorithm for the derivation described earlier. For conciseness, we assume that there is only one anchor and that is de[7]/that[7].

As shown, the algorithm starts with an empty stack. It then projects the source index to the corresponding target word and then enumerates the target string in a left to right fashion. If it finds a target word with a source index, it applies the shift operation, pushing the index to the stack. Unless the symbol corresponds to an anchor, it tries to apply the reduce operation. Line (4) indicates the special treatment to the anchor. If the symbol being read is a nonterminal, then we push the entire stack that corresponds to that nonterminal. For example, when the algorithm reads X_d at line (6), it pushes the entire stack from line (5).

As M_Ls and M_Rs are being incrementally constructed, we can immediately compute $P_{dom_o}(dom(a_m, a_n)|a_m, a_n)$ as soon as a partial derivation covers both a_m and a_n. For example, we can compute $P_{dom_1}(dom(\text{you}_5/\text{have}_8, \text{de}_7/\text{that}_7) = \quad)$, $P_{dom_1}(dom(\text{de}_7/\text{that}_7, \text{zhi}_{10}/\text{of}_4) = \quad)$ and $P_{dom_2}(dom(\text{you}_5/\text{have}_8, \text{zhi}_{10}/\text{of}_4) = \quad)$ at partial hypothesis $X_d \prec X_c$ which corresponds to a constituent spanning from 5-11.

7 Experiments

Our baseline systems is a state-of-the-art string-to-dependency system (Shen et al., 2008). The system is trained on 10 million parallel sentences that are available to the Phase 1 of the DARPA BOLT Chinese-English MT task. The training corpora include a mixed genre of newswire, weblog, broadcast news, broadcast conversation, discussion forums and comes from various sources such as LDC, HK Law, HK Hansard and UN data.

In total, our baseline model employs more than 50 features, including from our proposed dominance and orientation models. In addition to the standard

Model	newswire			weblog			newswire+weblog		
	BLEU	TER	Comb	BLEU	TER	Comb	BLEU	TER	Comb
	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
(1) S2D	37.63	53.17	7.77	27.60	57.19	14.77	33.39	54.97	10.79
(2) $+dom_1$	38.12	52.31	7.10	27.56	56.58	14.51	33.64	54.24	10.30
(3) $+dom_2$	38.31	52.28	6.99	27.66	56.57	14.45	33.78	54.20	10.21
(4) $+dom_3$	38.31	52.52	7.10	28.24	56.56	14.16	34.02	54.33	10.15
(5) $+dom_4$	*38.54*	*52.22*	*6.84*	28.38	56.55	14.08	*34.20*	*54.16*	*9.98*
(6) $+dom_5$	38.17	52.57	7.20	*28.67*	56.27	13.80	34.16	54.27	10.05
(7) $+dom_6$	38.17	52.52	7.18	28.64	*56.22*	*13.79*	34.10	54.18	10.04
(8) $+ori$	38.52	52.43	6.96	28.26	56.54	14.14	34.15	54.27	10.06
(9) $+ori+dom_1$	38.87	52.05	6.59	28.01	56.48	14.23	34.26	54.03	9.89
(10) $+ori+dom_2$	38.96	51.87	6.45	27.98	56.23	14.12	34.29	53.82	9.77
(11) $+ori+dom_3$	39.19	51.77	6.29	28.19	56.15	13.98	34.52	53.73	9.61
(12) $+ori+dom_4$	**39.34**	51.77	6.21	28.41	56.17	13.88	34.60	53.69	9.54
(13) $+ori+dom_5$	39.31	**51.67**	**6.18**	**28.62**	**56.09**	**13.74**	**34.76**	**53.65**	**9.45**

Table 3: The NIST MT08 results on newswire (nw), weblog (wb) and combined genres. S2D is the baseline string-to-dependency system (line 1). Lines 2-7 shows the results of the dominance model with $O = 1 - 6$. Line 8 shows result on adding ori to the baseline. Lines 9-13 shows the results of the orientation complemented with the dominance model with varying O. The best BLEU, TER and Comb on each genre of the first set are in *italic* while those of the second set are in **bold**. For BLEU, higher scores are better, while for TER and Comb, lower scores are better.

features such as translation probabilities, we incorporate features that are found useful for developing a state-of-the-art baseline, such as the provenance features (Chiang et al., 2011). We use a 6-gram language model, which was trained on 10 billion English words from multiple corpora, including the English side of our parallel corpus plus other corpora such as Gigaword (LDC2011T07) and Google News. We also train a class-based language model (Chen, 2009) on two million English sentences selected from the parallel corpus. As for our string-to-dependency system, we train 3-gram models for left and right dependencies and unigram for head using the target side of the parallel corpus. To train our models, we select a set of 5 million sentence pairs.

For the tuning and development sets, we set aside 1275 and 1239 sentences selected from LDC2010E30 corpus. We tune the feature weights with PRO (Hopkins and May, 2011) to minimize (TER-BLEU)/2 metric. As for the blind test set, we report the performance on the NIST MT08 evaluation set, which consists of 691 sentences from newswire and 666 sentences from weblog. We pick the weights that produce the highest development set scores to decode the test set.

We perform two sets of experiments. The first set looks at the contribution of the dominance model with varying values of o. The second one looks at the combination of the dominance model and the orientation model. Table 3 summarizes the experimental results on NIST MT08 sets, categorized by genres. We report the results on newswire genre in columns a-c, those on weblog genre in column d-f, and those on mixed genre in column g-i. The performance of our baseline string-to-dependency syntax-based SMT is shown in the first line.

Lines 2-7 in Table 3 show the results of our first set of experiments, starting from the result of dom_1, which looks at only at pairs of adjacent anchors, to the result of dom_6, which looks at pairs of anchors that are at most 5 anchors away. As shown in line 2, our dominance model provides a nice improvement of around 0.5 point over the baseline even if it only looks at restricted context. Increasing the order of our dominance model provides an additional gain. However, the gain is more pronounced in the weblog genre (up to around 1 BLEU point) than in the newswire genre. We conjecture that this may be the artifact of our tune set, which comes from the weblog genre. We stop at dom_6 because we observe that the weight of the feature score that corresponds to the maximum order ($o = 6$) has a negative sign, which often indicates a high correlation between the new features and existing ones.

Lines 8-13 in Table 3 shows the results of our second set of experiments. Line 8 shows the result of adding the orientation model (ori) to the baseline system. As shown, integrating ori shows a significant gain. On top of which, we then integrate dom_1 to dom_5. We see a very encouraging result as adding the dominance model increases the performance further, consistently over different value of o. This suggests that the dominance model is complementary to the orientation model. Our best result provides more than 1 BP improvement and 1 TER reduction consistently over different genres. We see this result as confirming our intuition that the global contextual information provided by our AG model can significantly improve the performance of SMT even in a state-of-the-art system.

8 Related Work

Our work intersects with existing work in many different respects. In this section, we mainly focus on work related to introducing higher-order contextual information to reordering model.

In providing global contextual information, our work is related to a large amount of literature. To name a few, Zens and Ney (2006) improves the lexicalized reordering model of Tillman (2004) by incorporating part-of-speech information. Chang et al. (2009) incorporates contexts from syntactic parse tree. Bach et al. (2009) exploits the dependency information and Xiong et al. (2012) uses the predicate-argument structure.

Vaswani et al. (2011) introduces rule markov models for a forest-to-string model in which the number of possible derivations is restricted. More recently, Durrani et al. (2013) and Zhang et al. (2013) cast reordering process as a Markov process. Similar to these models, our proposed model also provide context dependencies to the application of translation rules, however, as they focus on minimal translation units (MTU) where we focus on a selected set of translation units. (Banchs et al., 2005) introduces a bigram model for monotone phrase-based system, but their definition of translation units is suitable only for language pairs with limited reordering, such as translating Spanish to English.

In equating anchors with the function word class, our work is closely related to the function word-centered model of Setiawan et al. (2007), especially the orientation model. Our dominance model is closely related to the reordering model of Setiawan et al. (2009), except that they only look at pair of adjacent anchors, forming a chain structure instead of a graph like in our dominance model. Furthermore, we provide a discriminative treatment to the model to include a richer set of features including syntactic features. This work can be seen as modeling the identity of the neighboring of the anchors, similar to (Setiawan et al., 2013). However, instead of looking at the words at the borders, we look at whether the neighboring constituents contain other anchors.

9 Conclusion

We propose the “Anchor Graph” (AG) model to encode global contextual information. A selected set of translation units, which we call anchors, serves as the vertices of AG. And as the edges, we model two types of relations, namely the dominance and the precedence relations, where the former looks at the positions of the anchors in the derivation structure, while the latter looks at the positions of the anchors in the surface structure, resulting into two probabilistic models over edge labels. As the models look at the pairs of anchors that go beyond multiple translation units, our AG model provides global contextual information.

Our AG model embodies (admittedly crudely) some basic principles of sentence organization, namely categorization (in categorizing units into anchors and non-anchors), linear order (in modeling the precedence of anchors) and constituency structure (in modeling the dominance between anchors). We are encouraged by the facts that we learn these principles in an unsupervised way and that we can achieve a significant improvement over a strong baseline in a large-scale Chinese-to-English translation task. In the future, we hope to continue this line of research, perhaps by learning to identify anchors automatically from training data or by using our models to induce derivations directly from unaligned sentence pair.

Acknowledgements

We would like to acknowledge the support of DARPA under Grant HR0011-12-C-0015 for funding part of this work. The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the DARPA.

References

Nguyen Bach, Qin Gao, and Stephan Vogel. 2009. Source-side dependency tree reordering models with subtree movements and constraints. In *Proceedings of the Twelfth Machine Translation Summit (MTSummit-XII)*, Ottawa, Canada, August. International Association for Machine Translation.

Rafael E. Banchs, Josep M. Crego, Adrià de Gispert, Patrik Lambert, and José B. Mariño. 2005. Statistical machine translation of Euparl data by using bilingual n-grams. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 133–136, Ann Arbor, Michigan, June. 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 (SSST-3) at NAACL HLT 2009*, pages 51–59, Boulder, Colorado, June. Association for Computational Linguistics.

Stanley Chen. 2009. Shrinking exponential language models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 468–476, Boulder, Colorado, June. Association for Computational Linguistics.

Colin Cherry, Robert C. Moore, and Chris Quirk. 2012. On hierarchical re-ordering and permutation parsing for phrase-based decoding. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 200–209, Montréal, Canada, June. Association for Computational Linguistics.

David Chiang, Steve DeNeefe, and Michael Pust. 2011. Two easy improvements to lexical weighting. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 455–460, Portland, Oregon, USA, June. Association for Computational Linguistics.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Nadir Durrani, Alexander Fraser, and Helmut Schmid. 2013. Model with minimal translation units, but decode with phrases. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1–11, Atlanta, Georgia, June. Association for Computational Linguistics.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Yang Feng, Haitao Mi, Yang Liu, and Qun Liu. 2010. An efficient shift-reduce decoding algorithm for phrased-based machine translation. In *Coling 2010: Posters*, pages 285–293, Beijing, China, August. Coling 2010 Organizing Committee.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 848–856, Honolulu, Hawaii, October. Association for Computational Linguistics.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK., July. 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, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation, June.

Hendra Setiawan, Min-Yen Kan, and Haizhou Li. 2007. Ordering phrases with function words. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 712–719, Prague, Czech Republic, June. Association for Computational Linguistics.

Hendra Setiawan, Min Yen Kan, Haizhou Li, and Philip Resnik. 2009. Topological ordering of function words in hierarchical phrase-based translation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 324–332, Suntec, Singapore, August. Association for Computational Linguistics.

Hendra Setiawan, Bowen Zhou, Bing Xiang, and Libin Shen. 2013. Two-neighbor orientation model with cross-boundary global contexts. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages

1264–1274, Sofia, Bulgaria, August. Association for Computational Linguistics.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL-08: HLT*, pages 577–585, Columbus, Ohio, June. Association for Computational Linguistics.

Christoph Tillman. 2004. A unigram orientation model for statistical machine translation. In *HLT-NAACL 2004: Short Papers*, pages 101–104, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Ashish Vaswani, Haitao Mi, Liang Huang, and David Chiang. 2011. Rule markov models for fast tree-to-string translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 856–864, Portland, Oregon, USA, June. Association for Computational Linguistics.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–404, Sep.

Deyi Xiong, Min Zhang, and Haizhou Li. 2012. Modeling the translation of predicate-argument structure for smt. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 902–911, Jeju Island, Korea, July. Association for Computational Linguistics.

Richard Zens and Hermann Ney. 2006. Discriminative reordering models for statistical machine translation. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL): Proceedings of the Workshop on Statistical Machine Translation*, pages 55–63, New York City, NY, June. Association for Computational Linguistics.

Hui Zhang, Kristina Toutanova, Chris Quirk, and Jianfeng Gao. 2013. Beyond left-to-right: Multiple decomposition structures for smt. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 12–21, Atlanta, Georgia, June. Association for Computational Linguistics.