Quasi-Synchronous Grammars: Alignment by Soft Projection of Syntactic Dependencies

David A. Smith and **Jason Eisner**
Department of Computer Science
Center for Language and Speech Processing
Johns Hopkins University
Baltimore, MD 21218, USA
{dasmith,eisner}@jhu.edu

Abstract

Many syntactic models in machine translation are channels that transform one tree into another, or synchronous grammars that generate trees in parallel. We present a new model of the translation process: quasi-synchronous grammar (QG). Given a source-language parse tree T_1, a QG defines a *monolingual* grammar that generates translations of T_1. The trees T_2 allowed by this monolingual grammar are inspired by pieces of substructure in T_1 and aligned to T_1 at those points. We describe experiments learning quasi-synchronous context-free grammars from bitext. As with other monolingual language models, we evaluate the cross-entropy of QGs on unseen text and show that a better fit to bilingual data is achieved by allowing greater syntactic divergence. When evaluated on a word alignment task, QG matches standard baselines.

1 Motivation and Related Work

1.1 Sloppy Syntactic Alignment

This paper proposes a new type of syntax-based model for machine translation and alignment. The goal is to make use of syntactic formalisms, such as context-free grammar or tree-substitution grammar, without being overly constrained by them.

Let S_1 and S_2 denote the source and target sentences. We seek to model the conditional probability

$$p(T_2, A \mid T_1) \quad (1)$$

where T_1 is a parse tree for S_1, T_2 is a parse tree for S_2, and A is a node-to-node alignment between them. This model allows one to carry out a variety of alignment and decoding tasks. Given T_1, one can translate it by finding the T_2 and A that maximize (1). Given T_1 and T_2, one can align them by finding the A that maximizes (1) (equivalent to maximizing $p(A \mid T_2, T_1)$). Similarly, one can align S_1 and S_2 by finding the parses T_1 and T_2, and alignment A, that maximize $p(T_2, A \mid T_1) \cdot p(T_1 \mid S_1)$, where $p(T_1 \mid S_1)$ is given by a monolingual parser. We usually accomplish such maximizations by dynamic programming.

Equation (1) does not assume that T_1 and T_2 are isomorphic. For example, a model might judge T_2 and A to be likely, given T_1, provided that *many*—but not necessarily all—of the syntactic dependencies in T_1 are aligned with corresponding dependencies in T_2. Hwa et al. (2002) found that human translations from Chinese to English preserved only 39–42% of the unlabeled Chinese dependencies. They increased this figure to 67% by using more involved heuristics for aligning dependencies across these two languages. That suggests that (1) should be defined to consider more than one dependency at a time.

This inspires the key novel feature of our models: A does not have to be a "well-behaved" syntactic alignment. Any portion of T_2 can align to any portion of T_1, or to NULL. Nodes that are syntactically related in T_1 do *not* have to translate into nodes that are syntactically related in T_2—although (1) is usually higher if they do.

This property makes our approach especially promising for aligning freely, or erroneously, translated sentences, and for coping with syntactic **diver-**

Proceedings of the Workshop on Statistical Machine Translation, pages 23–30,
New York City, June 2006. ©2006 Association for Computational Linguistics

gences observed between even closely related languages (Dorr, 1994; Fox, 2002). We can patch together an alignment without accounting for all the details of the translation process. For instance, perhaps a source NP (figure 1) or PP (figure 2) appears "out of place" in the target sentence. A linguist might account for the position of the PP *auf diese Frage* either syntactically (by invoking scrambling) or semantically (by describing a deep analysis-transfer-synthesis process in the translator's head). But an MT researcher may not have the wherewithal to design, adequately train, and efficiently compute with "deep" accounts of this sort. Under our approach, it is possible to use a simple, tractable syntactic model, but with some contextual probability of "sloppy" transfer.

1.2 From Synchronous to Quasi-Synchronous Grammars

Because our approach will let anything align to anything, it is reminiscent of IBM Models 1–5 (Brown et al., 1993). It differs from the many approaches where (1) is defined by a stochastic synchronous grammar (Wu, 1997; Alshawi et al., 2000; Yamada and Knight, 2001; Eisner, 2003; Gildea, 2003; Melamed, 2004) and from transfer-based systems defined by context-free grammars (Lavie et al., 2003).

The synchronous grammar approach, originally due to Shieber and Schabes (1990), supposes that T_2 is generated in lockstep to T_1.[1] When choosing how to expand a certain VP node in T_2, a synchronous CFG process would observe that this node is aligned to a node VP′ in T_1, which had been expanded in T_1 by VP′ → NP′ V′. This might bias it toward choosing to expand the VP in T_2 as VP → V NP, with the new children V aligned to V′ and NP aligned to NP′. The process then continues recursively by choosing moves to expand these children.

One can regard this stochastic process as an instance of analysis-transfer-synthesis MT. Analysis chooses a parse T_1 given S_1. Transfer maps the context-free rules in T_1 to rules of T_2. Synthesis deterministically assembles the latter rules into an actual tree T_2 and reads off its yield S_2.

What is worrisome about the synchronous process is that it can only produce trees T_2 that are perfectly isomorphic to T_1. It is possible to relax this requirement by using synchronous grammar formalisms more sophisticated than CFG:[2] one can permit unaligned nodes (Yamada and Knight, 2001), duplicated children (Gildea, 2003)[3], or alignment between elementary trees of differing sizes rather than between single rules (Eisner, 2003; Ding and Palmer, 2005; Quirk et al., 2005). However, one would need rather powerful and slow grammar formalisms (Shieber and Schabes, 1990; Melamed et al., 2004), often with discontiguous constituents, to account for all the linguistic divergences that could arise from different movement patterns (scrambling, wh-*in situ*) or free translation. In particular, a synchronous grammar cannot practically allow S_2 to be any permutation of S_1, as IBM Models 1–5 do.

Our alternative is to define a "quasi-synchronous" stochastic process. It generates T_2 in a way that is not in thrall to T_1 but is "inspired by it." (A human translator might be imagined to behave similarly.) When choosing how to expand nodes of T_2, we are influenced both by the structure of T_1 and by monolingual preferences about the structure of T_2. Just as conditional Markov models can more easily incorporate global features than HMMs, we can look at the entire tree T_1 at every stage in generating T_2.

2 Quasi-Synchronous Grammar

Given an input S_1 or its parse T_1, a quasi-synchronous grammar (QG) constructs a monolingual grammar for parsing, or generating, the possible translations S_2—that is, a grammar for finding appropriate trees T_2. What ties this target-language grammar to the source-language input? The grammar provides for target-language words to take on

[1] The usual presentation describes a process that generates T_1 and T_2 jointly, leading to a joint model $p(T_2, A, T_1)$. Dividing by the marginal $p(T_1)$ gives a conditional model $p(T_2, A \mid T_1)$ as in (1). In the text, we directly describe an equivalent conditional process for generating T_2, A given T_1.

[2] When one moves beyond CFG, the derived trees T_1 and T_2 are still produced from a single derivation tree, but may be shaped differently from the derivation tree and from each other.

[3] For tree-to-tree alignment, Gildea proposed a *clone* operation that allowed subtrees of the source tree to be reused in generating a target tree. In order to preserve dynamic programming constraints, the identity of the cloned subtree is chosen independently of its insertion point. This breakage of monotonic tree alignment moves Gildea's alignment model from synchronous to quasi-synchronous.

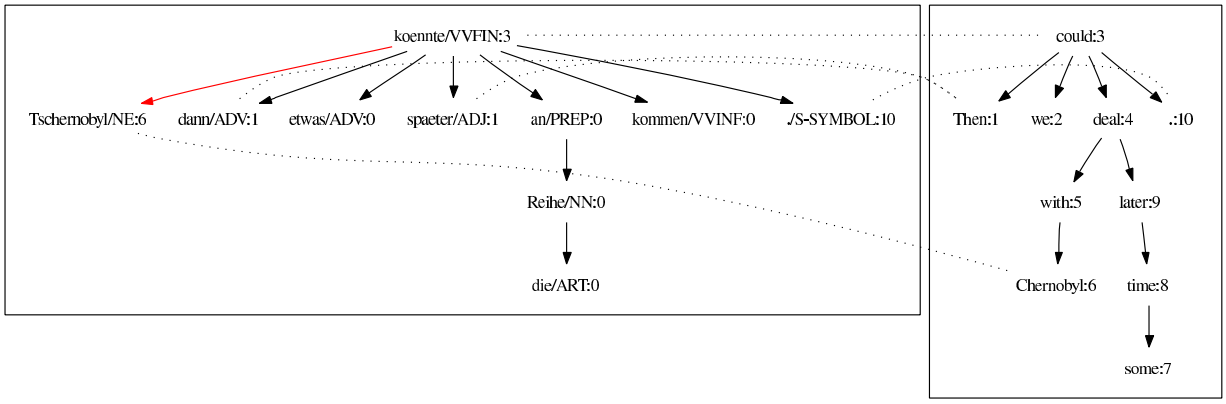

Figure 1: German and English dependency parses and their alignments from our system where German is the target language. *Tschernobyl* depends on *könnte* even though their English analogues are not in a dependency relationship. Note the parser's error in not attaching *etwas* to *später*.
German: *Tschernobyl könnte dann etwas später an die Reihe kommen .*
Literally: Chernobyl could then somewhat later on the queue come.
English: *Then we could deal with Chernobyl some time later .*

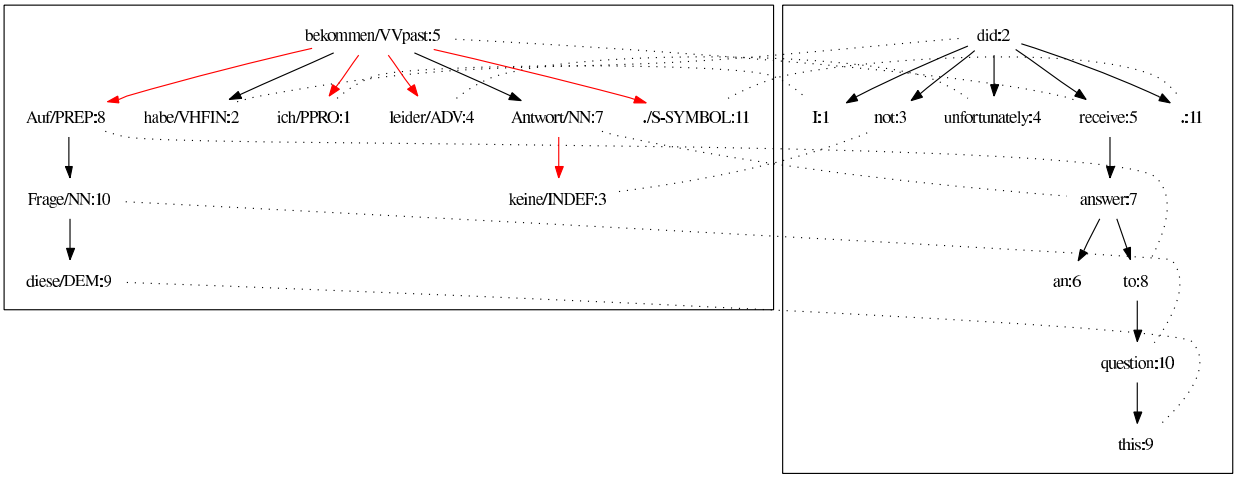

Figure 2: Here the German sentence exhibits scrambling of the phrase *auf diese Frage* and negates the object of *bekommen* instead of the verb itself.
German: *Auf diese Frage habe ich leider keine Antwort bekommen .*
Literally: To this question have I unfortunately no answer received.
English: *I did not unfortunately receive an answer to this question .*

multiple hidden "senses," which correspond to (possibly empty sets of) word tokens in S_1 or nodes in T_1. To take a familiar example, when parsing the English side of a French-English bitext, the word *bank* might have the sense *banque* (financial) in one sentence and *rive* (littoral) in another.

The QG[4] considers the "sense" of the former *bank* token to be a pointer to the particular *banque* token to which it aligns. Thus, a particular assignment of S_1 "senses" to word tokens in S_2 encodes a word alignment.

Now, selectional preferences in the monolingual grammar can be influenced by these T_1-specific senses. So they can encode preferences for how T_2 ought to copy the syntactic structure of T_1. For example, if T_1 contains the phrase *banque nationale*, then the QG for generating a corresponding T_2 may encourage any T_2 English noun whose sense is *banque* (more precisely, T_1's token of *banque*) to generate an adjectival English modifier with sense *nationale*. The exact probability of this, as well as the likely identity and position of that English modifier (e.g., *national bank*), may also be influenced by monolingual facts about English.

2.1 Definition

A quasi-synchronous grammar is a monolingual grammar that generates translations of a source-language sentence. Each state of this monolingual grammar is annotated with a "sense"—a set of zero or more nodes from the source tree or forest.

For example, consider a quasi-synchronous *context-free* grammar (QCFG) for generating translations of a source tree T_1. The QCFG generates the target sentence using nonterminals from the cross product $U \times 2^{V_1}$, where U is the set of monolingual target-language nonterminals such as NP, and V_1 is the set of nodes in T_1.

Thus, a binarized QCFG has rules of the form

$$\langle A, \alpha \rangle \rightarrow \langle B, \beta \rangle \langle C, \gamma \rangle \quad (2)$$

$$\langle A, \alpha \rangle \rightarrow w \quad (3)$$

where $A, B, C \in U$ are ordinary target-language nonterminals, $\alpha, \beta, \gamma \in 2^{V_1}$ are sets of source tree nodes to which A, B, C respectively align, and w is a target-language terminal.

Similarly, a quasi-synchronous tree-substitution grammar (QTSG) annotates the root and frontier nodes of its elementary trees with sets of source nodes from 2^{V_1}.

2.2 Taming Source Nodes

This simple proposal, however, presents two main difficulties. First, the number of possible senses for each target node is exponential in the number of source nodes. Second, note that the senses are sets of source tree nodes, not word types or absolute sentence positions as in some other translation models. Except in the case of identical source trees, source tree nodes will not recur between training and test.

To overcome the first problem, we want further restrictions on the set α in a QG state such as $\langle A, \alpha \rangle$. It should not be an *arbitrary* set of source nodes. In the experiments of this paper, we adopt the simplest option of requiring $|\alpha| \leq 1$. Thus each node in the target tree is aligned to a *single* node in the source tree, or to $\emptyset$ (the traditional NULL alignment). This allows one-to-many but not many-to-one alignments.

To allow many-to-many alignments, one could limit $|\alpha|$ to at most 2 or 3 source nodes, perhaps further requiring the 2 or 3 source nodes to fall in a particular configuration within the source tree, such as child-parent or child-parent-grandparent. With that configurational requirement, the number of possible senses α remains small—at most three times the number of source nodes.

We must also deal with the menagerie of different source tree nodes in different sentences. In other words, how can we tie the parameters of the different QGs that are used to generate translations of different source sentences? The answer is that the probability or weight of a rule such as (2) should depend on the specific nodes in α, β, and γ only through their properties—e.g., their nonterminal labels, their head words, and their grammatical relationship in the source tree. Such properties do recur between training and test.

For example, suppose for simplicity that $|\alpha| = |\beta| = |\gamma| = 1$. Then the rewrite probabilities of (2) and (3) could be log-linearly modeled using features that ask whether the single node in α has two children in the source tree; whether its children in the

[4] By abuse of terminology, we often use "QG" to refer to the T_1-specific monolingual grammar, although the QG is properly a recipe for constructing such a grammar from any input T_1.

source are the nodes in β and γ; whether its non-terminal label in the source is A; whether its fringe in the source translates as w; and so on. The model should also consider monolingual features of (2) and (3), evaluating in particular whether $A \rightarrow BC$ is likely in the target language.

Whether rule weights are given by factored generative models or by naive Bayes or log-linear models, we want to score QG productions with a small set of monolingual and bilingual features.

2.3 Synchronous Grammars Again

Finally, note that synchronous grammar is a special case of quasi-synchronous grammar. In the context-free case, a synchronous grammar restricts senses to single nodes in the source tree and the NULL node. Further, for any k-ary production

$$\langle X_0, \alpha_0 \rangle \rightarrow \qquad \langle X_1, \alpha_1 \rangle \ldots \langle X_k, \alpha_k \rangle$$

a synchronous context-free grammar requires that

1. $(\forall i \neq j)\ \alpha_i \neq \alpha_j$ unless $\alpha_i = \text{NULL}$,
2. $(\forall i > 0)\ \alpha_i$ is a child of α_0 in the source tree, unless $\alpha_i = \text{NULL}$.

Since NULL has no children in the source tree, these rules imply that the children of any node aligned to NULL are themselves aligned to NULL. The construction for synchronous tree-substitution and tree-adjoining grammars goes through similarly but operates on the derivation trees.

3 Parameterizing a QCFG

Recall that our goal is a conditional model of $p(T_2, A \mid T_1)$. For the remainder of this paper, we adopt a dependency-tree representation of T_1 and T_2. Each tree node represents a word of the sentence together with a part-of-speech tag. Syntactic dependencies in each tree are represented directly by the parent-child relationships.

Why this representation? First, it helps us concisely formulate a QG translation model where the source dependencies influence the generation of target dependencies (see figure 3). Second, for evaluation, it is trivial to obtain the word-to-word alignments from the node-to-node alignments. Third, the part-of-speech tags are useful backoff features, and in fact play a special role in our model below.

When stochastically generating a translation T_2, our quasi-synchronous generative process will be influenced by both fluency and adequacy. That is, it considers both the local well-formedness of T_2 (a monolingual criterion) and T_2's local faithfulness to T_1 (a bilingual criterion). We combine these in a simple generative model rather than a log-linear model. When generating the children of a node in T_2, the process first generates their tags using monolingual parameters (fluency), and then fills in in the words using bilingual parameters (adequacy) that select and translate words from T_1.[5]

Concretely, each node in T_2 is labeled by a triple (tag, word, aligned word). Given a parent node (p, h, h') in T_2, we wish to generate sequences of left and right child nodes, of the form (c, a, a').

Our *monolingual parameters* come from a simple generative model of syntax used for grammar induction: the Dependency Model with Valence (DMV) of Klein and Manning (2004). In scoring dependency attachments, DMV uses tags rather than words. The parameters of the model are:

1. $p_{choose}(c \mid p, dir)$: the probability of generating c as the next child tag in the sequence of dir children, where $dir \in \{left, right\}$.
2. $p_{stop}(s \mid h, dir, adj)$: the probability of generating no more child tags in the sequence of dir children. This is conditioned in part on the "adjacency" $adj \in \{true, false\}$, which indicates whether the sequence of dir children is empty so far.

Our *bilingual parameters* score word-to-word translation and aligned dependency configurations. We thus use the conditional probability $p_{trans}(a \mid a')$ that source word a', which may be NULL, translates as target word a. Finally, when a parent word h aligned to h' generates a child, we stochastically decide to align the child to a node a' in T_1 with one several possible relations to h'. A "monotonic" dependency alignment, for example, would have h' and a' in a parent-child relationship like their target-tree analogues. In different versions of the model, we allowed various dependency alignment configurations (figure 3). These configurations rep-

[5] This division of labor is somewhat artificial, and could be remedied in a log-linear model, Naive Bayes model, or deficient generative model that generates both tags and words conditioned on both monolingual and bilingual context.

resent cases where the parent-child dependency being generated by the QG in the target language maps onto source-language child-parent, for head swapping; the same source node, for two-to-one alignment; nodes that are siblings or in a c-command relationship, for scrambling and extraposition; or in a grandparent-grandchild relationship, e.g. when a preposition is inserted in the source language. We also allowed a "none-of-the-above" configuration, to account for extremely mismatched sentences.

The probability of the target-language dependency treelet rooted at h is thus:

$$P(D(h) \mid h, h', p) = \prod_{dir \in \{l,r\}} \prod_{c \in deps_D(p,dir)} P(D(c) \mid a, a', c) \times p_{stop}(nostop \mid p, dir, adj) \times p_{choose}(c \mid p, dir) \times p_{config}(config) \times p_{trans}(a \mid a') \; p_{stop}(stop \mid p, dir, adj)$$

4 Experiments

We claim that for modeling human-translated bitext, it is better to project syntax only loosely. To evaluate this claim, we train quasi-synchronous dependency grammars that allow progressively more divergence from monotonic tree alignment. We evaluate these models on cross-entropy over held-out data and on error rate in a word-alignment task.

One might doubt the use of dependency trees for alignment, since Gildea (2004) found that constituency trees aligned better. That experiment, however, aligned only the 1-best parse trees. We too will consider only the 1-best source tree T_1, but in constrast to Gildea, we will search for the target tree T_2 that aligns best with T_1. Finding T_2 and the alignment is simply a matter of parsing S_2 with the QG derived from T_1.

4.1 Data and Training

We performed our modeling experiments with the German-English portion of the Europarl European Parliament transcripts (Koehn, 2002). We obtained monolingual parse trees from the Stanford German and English parsers (Klein and Manning, 2003). Initial estimates of lexical translation probabilities came from the IBM Model 4 translation tables produced by GIZA++ (Brown et al., 1993; Och and Ney, 2003).

All text was lowercased and numbers of two or more digits were converted to an equal number of hash signs. The bitext was divided into training sets of 1K, 10K, and 100K sentence pairs. We held out one thousand sentences for evaluating the cross-entropy of the various models and hand-aligned 100 sentence pairs to evaluate alignment error rate (AER).

We trained the model parameters on bitext using the Expectation-Maximization (EM) algorithm. The T_1 tree is fully observed, but we parse the target language. As noted, the initial lexical translation probabilities came from IBM Model 4. We initialized the monolingual DMV parameters in one of two ways: using either simple tag co-occurrences as in (Klein and Manning, 2004) or "supervised" counts from the monolingual target-language parser. This latter initialization simulates the condition when one has a small amount of bitext but a larger amount of target data for language modeling. As with any monolingual grammar, we perform EM training with the Inside-Outside algorithm, computing inside probabilities with dynamic programming and outside probabilities through backpropagation.

Searching the full space of target-language dependency trees and alignments to the source tree consumed several seconds per sentence. During training, therefore, we constrained alignments to come from the union of GIZA++ Model 4 alignments. These constraints were applied only during training and not during evaluation of cross-entropy or AER.

4.2 Conditional Cross-Entropy of the Model

To test the explanatory power of our QCFG, we evaluated its conditional cross-entropy on held-out data (table 1). In other words, we measured how well a trained QCFG could predict the true translation of novel source sentences by summing over all parses of the target given the source. We trained QCFG models under different conditions of bitext size and parameter initialization. However, the principal independent variable was the set of dependency alignment configurations allowed.

From these cross-entropy results, it is clear that strictly synchronous grammar is unwise. We ob-

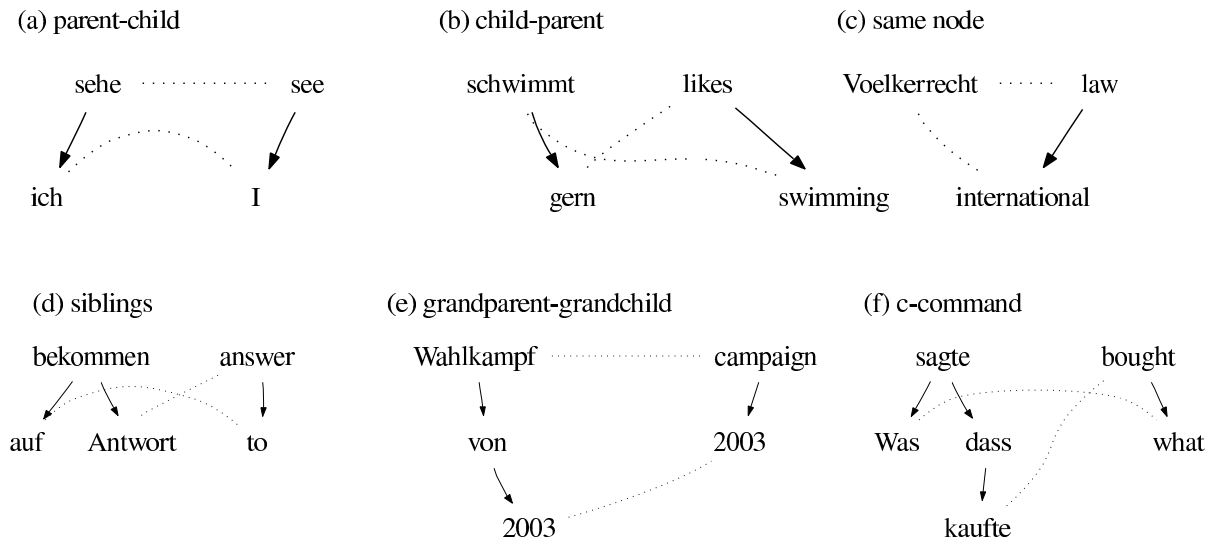

Figure 3: When a head h aligned to h' generates a new child a aligned to a' under the QCFG, h' and a' may be related in the source tree as, among other things, (a) parent–child, (b) child–parent, (c) identical nodes, (d) siblings, (e) grandparent–grandchild, (f) c-commander–c-commandee, (g) none of the above. Here German is the source and English is the target. Case (g), not pictured above, can be seen in figure 1, in English-German order, where the child-parent pair *Tschernobyl könnte* correspond to the words *Chernobyl* and *could*, respectively. Since *could* dominates *Chernobyl*, they are not in a c-command relationship.

Permitted configurations	CE at 1k	CE 10k	CE 100k
$\emptyset$ or parent-child (a)	43.82	22.40	13.44
+ child-parent (b)	41.27	21.73	12.62
+ same node (c)	41.01	21.50	12.38
+ all breakages (g)	35.63	18.72	11.27
+ siblings (d)	34.59	18.59	11.21
+ grandparent-grandchild (e)	34.52	**18.55**	**11.17**
+ c-command (f)	**34.46**	18.59	11.27
No alignments allowed	60.86	53.28	46.94

Table 1: Cross-entropy on held-out data with different dependency configurations (figure 3) allowed, for 1k, 10k, and 100k training sentences. The big error reductions arrive when we allow arbitrary non-local alignments in condition (g). Distinguishing some common cases of non-local alignments improves performance further. For comparison, we show cross-entropy when every target language node is unaligned.

tain comparatively poor performance if we require parent-child pairs in the target tree to align to parent-child pairs in the source (or to parent-NULL or NULL-NULL). Performance improves as we allow and distinguish more alignment configurations.

4.3 Word Alignment

We computed standard measures of alignment precision, recall, and error rate on a test set of 100 hand-aligned German sentence pairs with 1300 alignment links. As with many word-alignment evaluations, we do not score links to NULL. Just as for cross-entropy, we see that more permissive alignments lead to better performance (table 2).

Having selected the best system using the cross-entropy measurement, we compare its alignment error rate against the standard GIZA++ Model 4 baselines. As Figure 4 shows, our QCFG for German → English consistently produces better alignments than the Model 4 channel model for the same direction, German → English. This comparison is the appropriate one because both of these models are forced to align each English word to at most one German word. [6]

5 Conclusions

With quasi-synchronous grammars, we have presented a new approach to syntactic MT: constructing a monolingual target-language grammar that describes the aligned translations of a source-language sentence. We described a simple parameterization

[6]For German → English MT, one would use a German → English QCFG as above, but an English → German channel model. In this arguably inappropriate comparison, Figure 4 shows, the Model 4 channel model produces slightly better word alignments than the QG.

Permitted configurations	AER at 1k	AER 10k	AER 100k
$\emptyset$ or parent-child (a)	40.69	39.03	33.62
+ child-parent (b)	43.17	39.78	33.79
+ same node (c)	43.22	40.86	34.38
+ all breakages (g)	37.63	**30.51**	**25.99**
+ siblings (d)	37.87	33.36	29.27
+ grandparent-grandchild (e)	**36.78**	32.73	28.84
+ c-command (f)	37.04	33.51	27.45

Table 2: Alignment error rate (%) with different dependency configurations allowed.

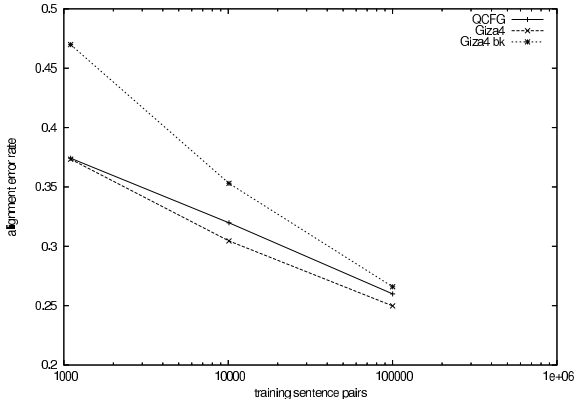

Figure 4: Alignment error rate with best model (all breakages). The QCFG consistently beat one GIZA++ model and was close to the other.

with gradually increasing syntactic domains of locality, and estimated those parameters on German-English bitext.

The QG formalism admits many more nuanced options for features than we have exploited. In particular, we now are exploring log-linear QGs that score overlapping elementary trees of T_2 while considering the syntactic configuration and lexical content of the T_1 nodes to which each elementary tree aligns.

Even simple QGs, however, turned out to do quite well. Our evaluation on a German-English word-alignment task showed them to be competitive with IBM model 4—consistently beating the German-English direction by several percentage points of alignment error rate and within 1% AER of the English-German direction. In particular, alignment accuracy benefited from allowing syntactic breakages between the two dependency structures.

We are also working on a translation decoding using QG. Our first system uses the QG to find optimal T_2 aligned to T_1 and then extracts a synchronous tree-substitution grammar from the aligned trees. Our second system searches a target-language vocabulary for the optimal T_2 given the input T_1.

Acknowledgements

This work was supported by a National Science Foundation Graduate Research Fellowship for the first author and by NSF Grant No. 0313193.

References

H. Alshawi, S. Bangalore, and S. Douglas. 2000. Learning dependency translation models as collections of finite state head transducers. *CL*, 26(1):45–60.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *CL*, 19(2):263–311.

Y. Ding and M. Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *ACL*, pages 541–548.

B. J. Dorr. 1994. Machine translation divergences: A formal description and proposed solution. *Computational Linguistics*, 20(4):597–633.

J. Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *ACL Companion Vol.*

H. J. Fox. 2002. Phrasal cohesion and statistical machine translation. In *EMNLP*, pages 392–399.

D. Gildea. 2003. Loosely tree-based alignment for machine translation. In *ACL*, pages 80–87.

D. Gildea. 2004. Dependencies vs. constituents for tree-based alignment. In *EMNLP*, pages 214–221.

R. Hwa, P. Resnik, A. Weinberg, and O. Kolak. 2002. Evaluating translational correspondence using annotation projection. In *ACL*.

D. Klein and C. D. Manning. 2003. Accurate unlexicalized parsing. In *ACL*, pages 423–430.

D. Klein and C. D. Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *ACL*, pages 479–486.

P. Koehn. 2002. Europarl: A multilingual corpus for evaluation of machine translation. http://www.iccs.informatics.ed.ac.uk/~pkoehn/-publications/europarl.ps.

A. Lavie, S. Vogel, L. Levin, E. Peterson, K. Probst, A. F. Llitjós, R. Reynolds, J. Carbonell, and R. Cohen. 2003. Experiments with a Hindi-to-English transfer-based MT system under a miserly data scenario. *ACM Transactions on Asian Language Information Processing*, 2(2):143 – 163.

I. D. Melamed, G. Satta, and B. Wellington. 2004. Generalized multitext grammars. In *ACL*, pages 661–668.

I. D. Melamed. 2004. Statistical machine translation by parsing. In *ACL*, pages 653–660.

F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *CL*, 29(1):19–51.

C. Quirk, A. Menezes, and C. Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *ACL*, pages 271–279.

S. M. Shieber and Y. Schabes. 1990. Synchronous tree-adjoining grammars. In *ACL*, pages 253–258.

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

K. Yamada and K. Knight. 2001. A syntax-based statistical translation model. In *ACL*.