Syntax Augmented Machine Translation via Chart Parsing with Integrated Language Modeling

Ashish Venugopal and **Andreas Zollmann**
School of Computer Science, Carnegie Mellon University, Pittsburgh
interACT Lab, University of Karlsruhe
{ashishv,zollmann}@cs.cmu.edu

Abstract

We present a hierarchical phrase-based translation model which annotates and generalizes existing phrase translations with syntactic categories derived from parsing the target side of a parallel corpus. We associate target parse trees for each training sentence pair with a search lattice constructed from the existing phrase translations on the corresponding source sentence, and consider techniques to produce a syntactically motivated bilingual synchronous grammar. We describe refinements to a chart based decoder and k-best extraction techniques to effectively parse the resulting grammar, which contains up to 4000 syntax-derivated nonterminals, producing translations that achieve significant improvements over Pharaoh, a state-of-the-art phrase based system, on the Europarl French-to-English task (Koehn and Monz, 2005).

1 Introduction

Recent work in machine translation has evolved from the traditional word (Brown et al., 1993) and phrase based (Koehn et al., 2004) models to include hierarchical phrase models (Chiang, 2005) and bilingual synchronous grammars (Melamed, 2004). These advances are motivated by the desire to integrate richer knowledge within the translation process to explicitly address limitations of the purely lexical phrase-based model. As (Chiang, 2005) and (Koehn et al., 2003) note, phrase-based models suffer from sparse data effects when required to translate conceptual elements that span or skip across several words, and distortion based reordering techniques tend to limit their range of operation for reasons of efficiency and model strength (Och and Ney, 2004).

Generalized phrases as discussed in (Chiang, 2005) and noted in (Block, 2000), attempt to directly address the limitiations of purely lexical phrases, and have shown significant improvements in translation quality by introducing constructs for sub-phrase representation. (Block, 2000) introduces a single generalization per phrase withing the EBMT framework, while (Chiang, 2005) can generate multiple generalizations within each phrase. In both these cases, however, generalizations are represented by a single sub-phrase category (and a glue rule for serial combination), providing the ability (and risk) of inserting any available sub-phrase into a larger phrase. To compete with state-of-the-art phrase based systems, (Chiang, 2005) extends this single category grammar by intersecting it with a n-gram language model, introducing additional nonterminal categories into the decoding process.

The formalism underlying hierarchical phrases is a synchronous context-free grammar (SynCFG), requiring a chart based decoding process that is significantly more computationally intensive than beam based decoding. Using a single generalization category X (left hand side in CFG notation) as in the work cited above, allows tractable parsing of the intersected grammar, at the cost of a more directed search process during parsing.

(Chiang, 2005) also restricts the grammar according to several noted principles, specifically allowing only 2 generalizations within a single rule and discarding rules which contain adjacent generalizations. These restrictions amongst others described are designed to compensate for the use of a single generalization category. It is easy to see

why they are necessary. Every phrase is marked with the same category X, allowing it to fill in any generalization of a phrase above it in the hierarchy. Without the knowledge of syntactic categories to restrict possible hierarchical combinations, these restrictions are required to make parsing tractable, at the expense of representational ability in the grammar.

In this work we consider the scenario where we have access to a target language parser to annotate and guide the generalization of the derived synchronous grammar. By associating target language parse trees with their corresponding search lattice built by lexical phrases (trained using traditional phrase extraction techniques (Koehn et al., 2004)) on the source sentence, we assign syntactic categories to phrases that align directly with the parse hierarchy. We also introduce syntax-derived categories that represent partially matched syntactic categories, thereby annotating every phrase in the initial phrase table. Our techniques produce around 4000 unique categories, limiting any attempt to intersect this grammar with a finite state n-gram language model. However, the resulting SynCFG also supports the translation process when the language model is confronted with unseen n-grams.

Our work addresses specific issues with inducing a grammar directly from parallel text, but does not move towards the work of (Yamada and Knight, 2002), where linguistic structures and motivation drive even the operation of the parsing process.

To accommodate for this extensive grammar, we introduce refinements to our chart based decoder and K-best extraction techniques that facilitate the use of a traditional n-gram language model during the decoding process. In this paper we will describe the generation of annotated and generalized phrases from traditional (lexical) phrase based resources, and develop techniques to efficiently parse the resulting synchronous grammar. We provide an approximation of the models described in (Chiang, 2005), and show that under the same parsing model, translation quality can be improved by considering syntactic and syntax-derived categories when generating the translation grammar.

2 Syntactic Synchronous Grammar

Traditional phrase-based translation as described in (Koehn et al., 2003) serves as the lexical foundation for our syntactic synchronous grammar (SynCFG)—syntactic, since its non-terminals are syntactic categories derived from parsing the target side of the parallel training corpus, and synchronous because they define operations to derive the source and target language simultaneously. Word alignment (Brown et al., 1993) driven phrase translations are extracted from the parallel training data, and we parse the target side of the training corpus with Charniak's parser (Charniak, 2000), set with parameters to generate parses quickly at the cost of some accuracy. With these resources (the phrase table and the target side parser for each sentence in the training data), we aim to construct a synchronous grammar of the form

$$X \rightarrow \langle \lambda, \alpha, \sim \rangle$$

to use the notation in (Chiang, 2005), where λ: $f_1 \ldots Y_i \ldots f_m$ is a sequence of terminals and non-terminals in the source language, α: $e_1 \ldots Y_j \ldots \ldots e_n$ is a sequence of terminals and non-terminals in the target language, and $\sim$ indicates a 1-1 correspondence between non-terminals Y across λ and α. Under this notation, phrase table entries define purely lexical λ and α.

To produce grammar rules from the phrase table entries, we annotate them with production categories and generalize them to form rules with non-terminals in their λ and α components by considering each sentence pair in the parallel corpus. For a given sentence pair, we apply the phrase table to the source sentence, creating a finite state lattice, that a traditional beam decoder would search to produce its translation output. We then consider the alignment of this lattice to the target side parse tree generated earlier, performing the annotation and generalization step as described below.

Annotation For each edge in the search lattice (i.e., target side of phrase pair), check if its span corresponds directly to a syntactic constituent in the target side parse tree. If we see an exact match, we assign the syntactic category to the phrase pair, setting the left hand side in the SynCFG rule. Phrase pairs that do not correspond to a span in the parse tree are given a default category "X", and can still play a role in the decoding process. In a variant of our rule extraction system, we as-

sign such phrases an extended category of the form $C_1 + C_2$, C_1/C_2, or $C_2\backslash C_1$, indicating that the phrase pair's target side spans two adjacent syntactic categories (e.g., *she went: NP+V*), a partial syntactic category C_1 missing a C_2 at the right (e.g., *the great: NP/NN*), or a partial C_1 missing a C_2 at the left (e.g., *great wall: DT\NP*), respectively.

Generalization In order to mitigate the effects of sparse data when working with phrase and n-gram models we would like to generate generalized phrases, which include non-terminal symbols that can be filled with other phrases. Therefore, after annotating the initial rules from the current training sentence pair, we adhere to (Chiang, 2005) to recursively generalize each existing rule

$$N \rightarrow f_1 \dots f_m / e_1 \dots e_n$$

for which there is an initial rule

$$M \rightarrow f_i \dots f_u / e_j \dots e_v$$

where $1 \leq i < u \leq m$ and $1 \leq j < v \leq n$, to obtain a new rule

$$N \rightarrow f_1 \dots f_{i-1} M_k f_{u+1} \dots f_m / e_1 \dots e_{j-1} M_k e_{v+1} \dots e_n$$

where k is a new index for the nonterminal M that expresses the one-to-one correspondence between the new occurrence of M on the source side and the corresponding one on the target side.

Unlike in (Chiang, 2005) we abstract on a per-sentence basis, and allow for different nonterminals in our grammar. Figures 1 illustrates the annotation and generalization process on the first sentence pair in the Europarl corpus.

Filtering Since our system allows for a vast number of possible rules to be extracted, we need to consolidate our rule base after around every 20000 training sentences processed. In this step, non-lexical rules that have occurred only once so far are eliminated from the rule bank.

Glue rule We also augment the grammar with a "glue" rule as per (Chiang, 2005) that allows us to connect partial derivations of the source sentence in series.

2.1 Decoding Features

As in (Chiang, 2005), we employ a log-linear model to decode a source sentence f with the SynCFG, representing translation quality in a set

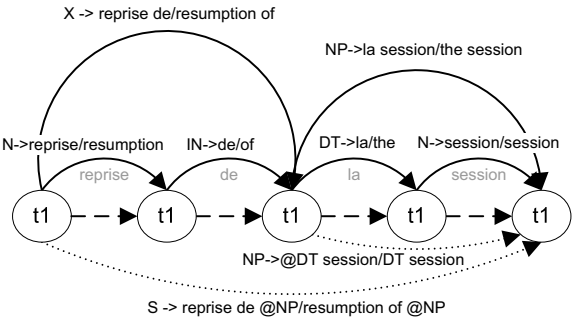

Figure 1: *Selected annotated and generalized (dotted arc) rules for the first sentence of the Europarl training corpus*

of features for each rule. We combine these rules with a n-gram language model to find a derivation $R(f)$ in a chart parsing decoder as described in the next section. We augment each grammar rule with the following features:

- source- and target-conditioned neg-log lexical weights as described in (Koehn et al., 2003)
- neg-log relative frequencies: left-hand-side-conditioned, target-phrase-conditioned, source-phrase-conditioned
- Counters: n.o. rule applications, n.o. target words
- Flags: IsPurelyLexical (i.e., contains only terminals), IsPurelyAbstract (i.e., contains only nonterminals), IsXRule (i.e., non-syntactical span), IsGlueRule
- Penalties: rareness penalty $\exp(1 - \textit{RuleFrequency})$ (highest penalty 1 if rule is a singleton, exponentially decaying with frequency); unbalancedness penalty $|\textit{MeanTargetSourceRatio} * \textit{'n.o. source words'} - \textit{'n.o. target words'}|$

3 Parsing

Given a source sentence f to translate with our SynCFG, we model the decoding process as a search through the derivation space of f, where the lowest cost derivation encodes a target translation

sequence in α. We define our translation model in log-linear space (dealing with costs, not scores) as

$$\arg \min_{R_0 \circ \cdots \circ R_n} lm(tgt_{R_0 \circ \cdots \circ R_n}) + \sum_{i=1}^{m} \lambda_i \sum_{j=1}^{n} (v^j)_i \ . \quad (1)$$

where $R_1 \circ \cdots \circ R_n$ is a derivation for f and $v^1, \ldots, v^n \in \mathbb{R}^m$ are the feature vectors of the applied rules $R_1, \ldots, R_n$. Further, $lm(tgt_{R_0 \circ \cdots \circ R_n})$ denotes the neg log probability of the target language sequence represented by the derivation $R_0 \circ \cdots \circ R_n$, and $\lambda_1, \ldots, \lambda_m$ are the parameters of the log-linear model, which are trained to maximize translation quality according to the BLEU metric (Papineni et al., 2002) on held out data using Minimum-Error-Rate training (Och, 2003).

The criterion defined above does not lend itself easily to a dynamic programming search of the derivation space due to the language model's dependance on decisions made at each subsequent step of the derivation. (Chiang, 2005) addresses this issue by intersecting his single-nonterminal grammar with an n-gram language model as a finite state automaton. In our work, this method is not appropriate, since each non-terminal symbol (from the set of up to 4000 in our extended-category variant) will become lexicalized with lexical information from the n-gram language model. A naive approximation would involve parsing without the language model and then rescoring.

Aside from the thorny issue of applying the language model during derivation search, we contend with a much larger search space than the traditional beam decoder. The SynCFG grammar extracted from the Europarl French-English corpus (details below) contains over 8 million rules after it has been filtered for the test set.

Instead of the common method of converting the CFG grammar into Chomsky Normal Form and applying a CYK algorithm to produce the most likely parse for a given source sentence, we avoided the explosion of the rule set caused by the introduction of new non-terminals in the conversion process and implemented the CYK+ algorithm (Chappelier and Rajman, 1998) in a variant that handles arbitrarily lexicalized rules.

Parsing framework In each chart cell we store a set of Nodes, each one corresponding to a particular syntactic category for that node. Within each Node, we maintain a heap of derivations, or completed hypotheses to use the CYK+ terminology, whose syntactic category matches the Node. Maintaining multiple Nodes in each cell orients our parser closer to a beam based parsing approach, since we cannot make definitive pruning decisions across derivations that result in difference categories during parsing. The syntactic categories serve a similar role to n-gram language models in a traditional beam based decoder. Within each Node is a minheap (fast access to the lowest cost element) of derivations. The lowest cost derivation (complete hypothesis for a span) serves as a prototype for this Node. Each complete hypothesis maintains a backwards star (Huang and Chiang, 2005) that points back to Nodes, rather than other complete hypotheses.

Matching Incomplete Rules The CYK+ algorithm relies on the ability to match incomplete parse hypotheses with complete hypotheses in a bottom up fashion. For any two cells in the chart (columns representing source sentence positions, and rows representing number of words spanned) that correspond to adjacent source spans, the CYK+ algorithm considers each combination of an incomplete hypothesis from one cell with a complete hypothesis in the other cell to apply the *Fundamental Rule*.

The incomplete hypotheses represent rules that are partially satisfied ('dotted rules') Most incomplete rules, however, will never actually be used in a successful derivation of the whole sentence. We store our rules in a BTree structure that allows quick addressing by prefixes of rule source-sides (λ), and only store prefixes (representing all incomplete hypotheses whose sequence of symbols left of the dot is equal to the stored prefix) within the chart cells. Thus, instead of storing and propagating incomplete hypotheses, we simply note that at least one rule has been been partially matched.

When we apply the Fundamental Rule, we query the prefix p concatenated with the production result r of the complete hypothesis (that we are trying to combine with the virtual incomplete rule) against the rule source sides in our BTree. If an exact match occurs, this implies that we have found a complete rule whose span is properly represented in our chart and we can form a new complete hypothesis. If we have a prefix match, we store pr as a new prefix in the corresponding chart cell, standing for all incomplete hypotheses with pr left of the dot.

Derivation Depth When considering hierarchical rules, we often encounter multiple derivations that generate identical target sequences, adding considerably to the number of hypotheses considered as we move up the chart during the CYK+ process.

We introduce a pruning parameter that restricts the use of partial derivations in further calls to *Fundamental Rule*. We want to only expand partial derivations that have not engaged in redundant rule usage, i.e., we want to introduce a preference towards short derivations. Therefore, when filing hypotheses in a chart cell of source length i and source position j, before executing the inner loop of the CYK algorithm, we determine in a look-ahead step the minimum number MinRuleAppCount(i,j) of rule applications possible across all combinations of complete hypotheses in cell (source length, source start)=(k, j) with incomplete hypotheses in cell $(i - k, j + k)$. In the actual k-loop, we now only allow those derivations whose number of rules applied is within a parameter *MaxRuleAppCountDifference* of the minimum number MinRuleAppCount(i,j) possible for this cell (i,j). While this pruning is greedy, we find its impact on translation scores minimal in practice.

Parsing Timeout During the decoding process, the parser has access to "glue" rules, that allow two derivations to be joined if they have adjacent spans in the source sentence. These derivations compete with derivations that have applied generalized rule forms. Minimally, glue rules and the cell word-to-word translation rules are the only rules required to obtain a complete derivation. Given the nature of our rule generation process, the number of derivations considered during parsing, especially for long sentences, can be prohibitively large. We introduce an additional pruning parameter *MaxCombinationCount* that limits the number of hierarchical rule applications that can be performed while parsing a sentence. Once this limit has been reached, only glue rules can be applied. This parameter effectively serves as a timeout, falling back on the "glue" rule to generate a full sentence parse. In our experiments, we set *MaxCombinationCount* $= 250000$, resulting in the generation of non-glue complete hypotheses for chart cells of maximum length between 7 and 13 (depending on the length of the test sentence—smaller maximum lengths for long test sentences, greater maximum lengths for short test sentences since there are less cells of a given length to be explored).

Lazier-than-Lazy We propose a parsing solution that uses weak estimates of the language model during the parsing process, and then stronger estimates of the language model during our K-best retrieval, culminating in exact language model scores assigned to all elements in the K-best lists.

During the parsing process, when a complete hypothesis is formed (derivation for a source side span), we immediately estimate a language model probability for that derivation. We use a Viterbi approximation to identify the target words of this derivation, considering the backward star Nodes and their prototypical hypothesis for their target word derivations. We perform this process recursively as we parse through the chart so that we do not have to re-evaluate the whole derivation each time. If the target words for a derivation are $e_1 \dots e_l$, we compute $p(e_l|e_{l-1} \dots e_{l-c}$ under the standard independence assumptions in the n-gram model with context size c. The first c words and and the last c words are noted in this derivation, and this derivation is added to its corresponding Node (based on its syntactic category). Its cost for the min-heap is calculated using the equation in 1, with the language model component being approximated by the estimate that we described here.

During K-Best retrieval, we adopt a strategy similar to (Huang and Chiang, 2005), but add an additional level of lazy management. Instead of expanding breadth-first, we expand depth-first, avoiding the risk of adding several alternative hypotheses to the beam that may not factor in the Top-K after the language model has been applied. For example, if the final sentence spanning derivation used the rule "$S \rightarrow$ I @VP to @NP", (@ is used to indicate non-terminal categories), then we would expand $@VP$ first, adding for example "$S \rightarrow$ I went to NP", and "$S \rightarrow$ I left to NP", to the search beam, instead of directly adding '$S \rightarrow$ I left to the house". Complete derivations are stored on the beam by equation 1, during lazier K-Best retrieval we update the language model estimate of each derivation that we add onto the beam. Since this K-Best variant expands derivations from left to right, we can correct the language model estimate as we go, until we finally have the exact language model estimate for the sentence spanning

derivation.

Unique Derivations The nature of our extracted SynCFG implies that there will be several derivations that result in the same target words being generated, leading to limited diversity in the K-Best list which is used for Minimum Error Rate (Och, 2003) optimization. We address this issue during our lazier K-Best list retrieval. When a derivation "pNewHyp" (which will contain terminals and non-terminals) is about to be added to the K-Best retrieval beam, we check to see if there has already been another derivation "pExistingHyp" that generated the same target words *and* refers to the same non-terminals, that has already been added to the beam. If the total cost of "pNewHyp" is higher than "pExistingHyp" then we avoid adding "pNewHyp" to the beam. Introducing this feature dramatically improves the diversity in the final K-Best list. Before applying this pruning in the beam, requesting a 1000 best list for one sentence would usually yield approximately 100 unique translations. After applying this pruning we consistently retrieve around 1750 unique translations on average when we set K to 2000.

4 Results

We pesent experiments on the Europarl French-English task as defined at the NAACL 2006 workshop: Exploiting Parallel Texts for Statistical Machine Translations. We compare a state-of-the-art phrase-based system against several degrees of modeling refinement within our system. All systems use the same initial phrase table (maximum phrase length 7) generated by the scripts provided for the workshop described in (Koehn et al., 2003). The language model is also provided in the 2006 shared task, and is built on 13 million English words using Knesser-Ney smoothing. We evaluated our results using the BLEU metric (Papineni et al., 2002), optimizing the parameters on the first 500 sentences of the provided 'Development Set' (identical to last year's development set), and testing on the provided 'Development Test Set' (identical to last year's test set). The threshold for statistical significance is 0.78 BLEU points at the 95 percent confidence level as calculated by (Zhang and Vogel, 2005).

The baseline phrase based translation system is Pharaoh (Koehn et al., 2004), using the default settings specified by the provided minimum-error-rate training scripts (phrase pruning b=100, chart pruning = 1e-5, distortion limit=4, K-Best=100). Minimum Error Rate training is run for 13 iterations till convergence, compensating for the relatively smaller K-Best size compared to our experiments.

Our systems are trained for two MER iterations and run with *MaxRuleAppCountDifference=1*, *MaxCombinationCount=250000*, and K-Best=2000.

- Baseline - Pharaoh as described above
- Lex - Phrase-decoder simulation: using only the initial lexical rules from the phrase table, all with LHS X, and the glue rule. An additional re-ordering rule is added for swap based re-ordering and a feature is added to reflect this operation (making it comparable to traditional phrase+reordering systems).
- XCat - All nonterminals are merged into a single X nonterminal - identical filtering to (Chiang, 2005)
- Syn - Syntactic extraction using the Penn Treebank parse categories as nonterminals; rules containing up to 4 nonterminal abstraction sites.
- SynExt - Syntactic extraction using the extended-category scheme, but with rules only containing up to 3 nonterminal abstraction sites.

We also explored the impact of longer initial phrases by training another phrase table with phrases up to length 12. The results based on the length-7 phrase table as well as the length-12 phrase table are presented in Table 1.

Our preliminary results show a statistically significant improvement of the Syn and SynExt system over the traditional phrase based decoding system. We also see a clear trend towards improving translation quality as we employ richer extraction techniques. However, our results do not show as great an improvement over the baseline as (Chiang, 2005) reported on the Chinese-English Tides data. We believe that this is due to the difference in language pairs, French offers less opportunities to benefit from stronger and better informed reordering models. We expect that before the final version of this paper we will also have results on

System	N.o. nonterminals	DevSet BLEU	TestSet BLEU
Baseline - max. phrase length 7	0	31.11	30.61
Lex - max. phrase length 7	2	28.96	29.12
XCat - max. phrase length 7	2	30.89	31.01
Syn - max. phrase length 7	75	31.52	31.31
SynExt - max. phrase length 7	3900	31.73	31.41
Baseline - max. phr. length 12	0	31.16	30.90
Lex - max. phr. length 12	2	29.30	29.51
XCat - max. phr. length 12	2	30.79	30.59
SynExt - max. phr. length 12	3900	31.07	31.76

Table 1: Translation results (IBM BLEU) for each system on the Fr-En '06 Shared Task 'Development Set' (used for MER parameter tuning) and '06 'Development Test Set' (identical to last year's Shared Task's test set).

Arabic and Chinese, languages where re-ordering plays a more significant role.

Note also that our decoding performance with the basic Lex system (which is essentially phrase based) is significantly below par compared to direct beam based decoding. As we continue to improve the integration between the language model and the decoder we expect to see improvements of this baseline as well, with the effect of improving the performance on each consecutive method.

4.1 Conclusions

In this work we applied syntax based resources (the target language parser) to annotate and generalize phrase translation tables extracted via existing phrase extraction techniques. Our work affirms the feasibility of parsing approaches to machine translation in a large data setting, while still taking advantage of a n-gram language model to assist the parsing process. We illustrated the impact of adding syntactic categories to drive and constrain the structured search space and to play a complementary role to the traditional language modeling approach. We expect our further work to involve experiments with languages where reordering effects are more prominent, allowing this syntax based approach to have a more significant impact on translation quality.

Our contributions to the integration of an n-gram language modeling component within the parsing process in the form of optimistic estimation during parsing, lazier K-Best retrieval and forcing unique translations within the K-Best process can have a significant impact on the state-of-the-art in the emerging hierarchical parsing domain. Unique K-Best lists are critically important to effective search space exploration and optimization of model parameters, and we expect to continue our work to more tightly integrate the language model during the parsing process.

Our translation system is available open-source under the GNU General Public License at: `www.cs.cmu.edu/~zollmann/samt`

5 Acknowledgments

This work has been partly funded by the European Union under the integrated project TC-STAR - Technology and Corpora for Speech to Speech Translation (IST-2002-FP6-506738, http://www.tc-star.org).

References

Hans Ulrich Block. 2000. Example based incremental synchronous interpretation. In *Vermobil: Foundations of Speech-to-Speech Translation*.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19(2):263–311.

J.-C. Chappelier and M. Rajman. 1998. A generalized CYK algorithm for parsing stochastic CFG. In *Proceedings of Tabulation in Parsing and Deduction (TAPD'98)*, pages 133–137, Paris.

Eugene Charniak. 2000. A maximum entropy-inspired parser. In *Proceedings of the North American Association for Computational Linguistics (HLT/NAACL)*.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. of the Association for Computational Linguistics*.

Liang Huang and David Chiang. 2005. Better k-best parsing. In *Proceedings of the 9th International Workshop on Parsing Technologies*.

Phillip Koehn and Christof Monz. 2005. Shared task: Statistical machine translation between european languages. In *Proceedings of the Association of Computational Linguistics Workshop on Parallel Corpora*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, Edomonton, Canada, May 27-June 1.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2004. Pharaoh: A beam search decoder for phrase-base statistical machine translation models. In *Proceedings of the Sixth Confernence of the Association for Machine Translation in the Americas*, Edomonton, Canada, May 27-June 1.

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

Franz Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Comput. Linguist.*

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of the Association for Computational Linguistics*, Sapporo, Japan, July 6-7.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the Association of Computational Linguistics*, pages 311–318.

Kenji Yamada and Kevin Knight. 2002. A decoder for syntax-based statistical mt. In *Proc. of the Association for Computational Linguistics*.

Ying Zhang and Stephan Vogel. 2005. An efficient phrase-to-phrase alignment model for arbitrarily long phrase and large corpora. In *Proceedings of the Tenth Conference of the European Association for Machine Translation (EAMT-05)*, Budapest, Hungary, May. The European Association for Machine Translation.