The HiFST System for the EuroParl Spanish-to-English Task *

Gonzalo Iglesias[⋆] **Adrià de Gispert**[‡] **Eduardo R. Banga**[⋆] **William Byrne**[‡]
[⋆] University of Vigo. Dept. of Signal Processing and Communications. Vigo, Spain
{giglesia,erbanga}@gts.tsc.uvigo.es
[‡] University of Cambridge. Dept. of Engineering. CB2 1PZ Cambridge, U.K.
{ad465,wjb31}@eng.cam.ac.uk

Abstract: In this paper we present results for the Europarl Spanish-to-English translation task. We use HiFST, a novel hierarchical phrase-based translation system implemented with finite-state technology that creates target lattices rather than k-best lists.
Keywords: Statistical Machine Translation, Hierarchical Phrase-Based Translation, Rule Patterns

1 Introduction

In this paper we report translation results in the EuroParl Spanish-to-English translation task using HiFST, a novel hierarchical phrase-based translation system that builds word translation lattices guided by a CYK parser based on a synchronous context-free grammar induced from automatic word alignments. HiFST has achieved state-of-the-art results for Arabic-to-English and Chinese-to-English translation tasks (Iglesias et al., 2009a). This decoder is easily implemented with Weighted Finite-State Transducers (WFSTs) by means of standard operations such as union, concatenation, epsilon removal, determinization, minimization and shortestpath, available for instance in the OpenFst libraries (Allauzen et al., 2007). In every CYK cell we build a single, minimal word lattice containing all possible translations of the source sentence span covered by that cell. When derivations contain non-terminals, arcs refer to lower-level lattices for memory efficiency, working effectively as pointers. These are only expanded to the actual translations if pruning is required during search; expansion is otherwise only carried out at the upper-most cell, after the full CYK grid has been traversed.

* This work was supported in part by the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011- 06-C-0022. G. Iglesias supported by Spanish Government research grant BES-2007-15956 (project TEC2006-13694-C03-03).

1.1 Related Work

Hierarchical translation systems (Chiang, 2007), or simply Hiero, share many common features with standard phrase-based systems (Koehn, Och, and Marcu, 2003), such as feature-based translation and strong target language models. However, they differ greatly in that they guide their reordering model by a powerful synchronous context-free grammar, which leads to flexible and highly lexicalized reorderings. They yield state-of-the-art performance for some of the most complex translation tasks, such as Chinese-to-English. Thus, Hiero decoding is one of the dominant trends in the field of Statistical Machine Translation.

We summarize some extensions to the basic approach to put our work in context.

Hiero Search Refinements: Huang and Chiang (2007) offer several refinements to cube pruning to improve translation speed. Venugopal et al. (2007) introduce a Hiero variant with relaxed constraints for hypothesis recombination during parsing; speed and results are comparable to those of cube pruning, as described by Chiang (2007). Li and Khudanpur (2008) report significant improvements in translation speed by taking unseen n-grams into account within

cube pruning to minimize language model requests. Dyer et al. (2008) extend the translation of source sentences to translation of input lattices following Chappelier et al. (1999).

Extensions to Hiero: Several authors describe extensions to Hiero, to incorporate additional syntactic information (Zollmann and Venugopal, 2006; Zhang and Gildea, 2006; Shen, Xu, and Weischedel, 2008; Marton and Resnik, 2008), or to combine it with discriminative latent models (Blunsom, Cohn, and Osborne, 2008).

Analysis and Contrastive Experiments: Zollman et al. (2008) compare phrase-based, hierarchical and syntax-augmented decoders for translation of Arabic, Chinese, and Urdu into English. Lopez (2008) explores whether lexical reordering or the phrase discontiguity inherent in hierarchical rules explains improvements over phrase-based systems. Hierarchical translation has also been used to great effect in combination with other translation architectures, e.g. (Sim et al., 2007; Rosti et al., 2007).

WFSTs for Translation: There is extensive work in using Weighted Finite State Transducers for machine translation (Bangalore and Riccardi, 2001; Casacuberta, 2001; Kumar and Byrne, 2005; Mathias and Byrne, 2006; Graehl, Knight, and May, 2008).

This paper proceeds as follows. Section 2 describes the HiFST system; section 3 provides details of the translation task and optimization. Section 4 discusses different strategies in order to reduce the size of grammar. Finally, section 5 shows rescoring results, after which we conclude.

2 HiFST System Description

In brief, HiFST is a hierarchical decoder that builds target word lattices guided by a synchronous context-free grammar consisting of a set $\mathbf{R} = \{R^r\}$ of rules $R^r : N \rightarrow \langle \gamma^r, \alpha^r \rangle$ / p^r. A priori, N represents any non-terminal; in this paper, N can be either S, X, or V. As usual, the special glue rules $S \rightarrow \langle X, X \rangle$ and $S \rightarrow \langle S\ X, S\ X \rangle$ are included. $\mathbf{T}$ denotes the terminals (words), and the grammar builds parses based on strings $\gamma, \alpha \in \{\{S, X, V\} \cup \mathbf{T}\}^+$, where we follow Chiang's general restrictions to the grammar (2007).

As shown in Figure 1, the system performs translation in three main steps. The first step is a variant of the classic Cocke-Younger-Kasami (CYK) algorithm closely related to CYK+ (Chappelier and Rajman, 1998), for which hypothesis recombination without pruning is performed and back-pointers are mantained. Although the model is a synchronous grammar, in this stage only the source language sentence is parsed using the corresponding context-free grammar with rules $N \rightarrow \gamma$. Each cell in the CYK grid is specified by a non-terminal symbol and position in the CYK grid: (N, x, y), which spans s_x^{x+y-1} on the source sentence $s_1 ... s_J$.

For the second step, we use a recursive algorithm to construct word lattices with all possible translations produced by the hierarchical rules. Construction proceeds by traversing the CYK grid along the back-pointers established in parsing. In each cell (N, x, y) of the CYK grid, we build a target language word lattice $\mathcal{L}(N, x, y)$. This lattice contains every translation of s_x^{x+y-1} from every derivation headed by N. For each rule in this cell, a simple acceptor is built based on the target side, as the result of standard concatenation of acceptors that encode either terminals or non-terminals. A word is encoded trivially with an acceptor of two states binded by a single transition arc. In turn, a non-terminal corresponds to a lattice retrieved by means of its back-pointer to a low-level cell. If this low-level lattice has not been required previously, it has to be built first. Once we have all the acceptors (one per rule) that apply to (N, x, y), we obtain the cell lattice $\mathcal{L}(N, x, y)$ by unioning all these acceptors. For a source sentence with J words, the lattice we are looking for is at the top cell $(S, 1, J)$.

The final translation lattice $\mathcal{L}(S, 1, J)$ can grow very large after the pointer arcs are expanded. Therefore, in the third step we apply a word-based language model via WFST composition, and perform likelihood-based pruning (Allauzen et al., 2007) based on the combined translation and language model scores.

This method can be seen as a generalization of the k-best algorithm with cube pruning (Chiang, 2007), as the performance of this cube pruning search is clearly limited by the size k of each k-best list. For small tasks where k is sufficiently large compared to the number of translations of each derivation, search could be exhaustive. On the other hand, for reasonably large tasks, the inven-

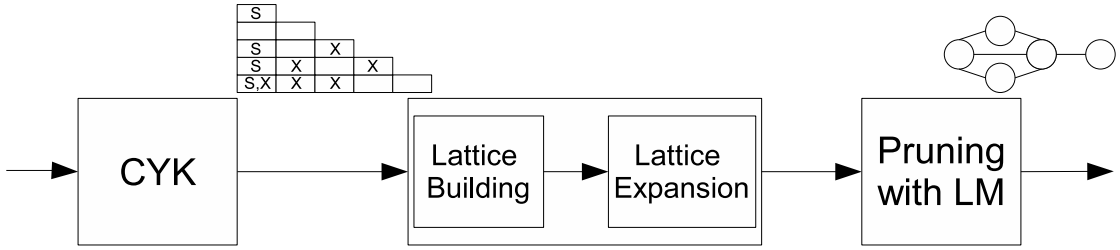

Figure 1: The HiFST System

tory of hierarchical phrases is much bigger than standard phrase pair tables, and using a large k is impossible without exponentially increasing memory requirements and decoding time. In practice, values of (no more than) k=1000 or k=10000 are used. This results in search errors. Search errors have two negative consequences. Clearly, translation quality is undermined as the obtained first-best hypothesis is suboptimal given the models. Additionally, the quality of the obtained k-best list is also suboptimal, which limits the margin of gain potentially achieved by subsequent re-ranking strategies (such as high order language model rescoring or Minimum Bayes Risk).

2.1 Skeleton Lattices

As explained previously, the lattice building step uses a recursive algorithm. If this is actually carried out over full word lattices, memory consumption would increase dramatically. Fortunately, we can avoid this by using arcs that encode a reference to low level lattices, thus working as pointers. In other words, we actually build for each cell a skeleton of the desired lattice. This skeleton lattice is a mixture of words and pointers to low-level cell lattices, but WFST operations are still possible. In this way, skeleton lattices in each cell can be greatly reduced through operations such as epsilon removal, determinization and minimization. Expansion is ideally carried out with the final skeleton lattice in $(S, 1, J)$ using a single recursive replacement operation. Nevertheless, redundancy still remains in the final expanded lattice as concatenation and union of sublattices with different spans typically lead to repeated hypotheses.

The use of the lattice pointer arc was inspired by the 'lazy evaluation' techniques developed by Mohri et al (2000). Its implementation uses the infrastructure provided by the OpenFST libraries for delayed composition, etc.

	sentences	*words*	*vocab*
ES	1.30M	38.2M	140k
EN		35.7M	106k

Table 1: Parallel corpora statistics.

2.2 Pruning in Lattice Building

Ideally, pruning-in-search performed during lattice building should be avoided, but this is not always possible, depending on the complexity of the grammar and the size of the sentence. In HiFST this pruning-in-search is triggered selectively by three concurrent factors: number of states of a given lattice in a cell, size of source-side span (in words) and the non-terminal category. This pruning-in-search consists of applying the language model via composition, likelihood-based pruning and posterior removal of language model scores.

3 Development and Tuning

We present results on the Spanish-to-English translation shared task of the ACL 2008 Workshop on Statistical Machine Translation, WMT (Callison-Burch et al., 2008). The parallel corpus statistics are summarized in Table 1. Specifically, throughout all our experiments we use the Europarl *dev2006* and *test2008* sets for development and test, respectively.

The training was performed using lowercased data. Word alignments were generated using GIZA++ (Och, 2003) over a stemmed[1] version of the parallel text. After unioning the Viterbi alignments, the stems were replaced with their original words, and phrase-based rules of up to five source words in length were extracted (Koehn, Och, and

[1]We used snowball stemmer, available at http://snowball.tartarus.org.

Marcu, 2003). Hierarchical rules with up to two non-contiguous non-terminals in the source side are then extracted applying the restrictions described by Chiang (2007).

The Europarl language model is a Kneser-Ney (Kneser and Ney, 1995) smoothed default cutoff 4-gram back-off language model estimated over the concatenation of the Europarl and News language model training data.

Minimum error training (Och, 2003) under BLEU (Papineni et al., 2001) is used to optimize the feature weights of the decoder with respect to the *dev2006* development set. We obtain a k-best list from the translation lattice and extract each feature score with an aligner variant of a k-best cube-pruning decoder. This variant produces very efficiently the most probable rule segmentation that generated the output hypothesis, along with each feature contribution. The following features are optimized:

- Target language model
- Number of usages of the glue rule
- Word and rule insertion penalties
- Word deletion scale factor
- Source-to-target and target-to-source translation models
- Source-to-target and target-to-source lexical models
- Three rule count features inspired by (Bender et al., 2007) that classify rule ocurrences (one, two, or more than two times respectively).

4 Grammar Design

In order to work with reasonably small grammar – yet competitive in performance, we apply three filtering strategies succesfully used for Chinese-to-English and Arabic-to-English translation tasks (Iglesias et al., 2009b).

- Pattern and mincount-per-class filtering
- Hiero Shallow model
- Filtering by number of translations.

We also add deletion rules, i.e. rules that delete single words. In the following subsections we explain these strategies.

Excluded Rules	Types
$\langle X_1 w, X_1 w \rangle$, $\langle w X_1, w X_1 \rangle$	1530797
$\langle X_1 w X_2, * \rangle$	737024
$\langle X_1 w X_2 w, X_1 w X_2 w \rangle$, $\langle w X_1 w X_2, w X_1 w X_2 \rangle$	41600246
$\langle w X_1 w X_2 w, * \rangle$	45162093
$N_{nt}.N_e$= 1.3 mincount=5	39013887
$N_{nt}.N_e$= 2.4 mincount=10	6836855

Table 2: Rules excluded from grammar G.

4.1 Filtering by Patterns and Mincounts

Even after applying the rule extraction constraints proposed by Chiang (2007), our initial grammar G for *dev2006* exceeds 138M rules, of which only 1M are simple phrase-based rules. With the following procedure we reduce the size of the initial grammar with a more informed criterion than general mincount filtering.

A rule pattern is simply obtained by replacing every sequence of terminals by a single symbol 'w' (indicating word, i.e. terminal string, $w \in \mathbf{T}^+$). Every rule of the grammar has one unique corresponding rule pattern. For instance, rule:

$$\langle \text{ X}_2 \text{ en X}_1 \text{ ocasiones , on X}_1 \text{ occasions X}_2 \rangle$$

would correspond to the rule pattern:

$$\langle \text{ X}_2\text{w X}_1\text{w , w X}_1\text{w X}_2 \rangle$$

We further classify patterns by number of non-terminals N_{nt} and elements N_e (non-terminals and substring of terminals). There are 5 possible classes: $N_{nt}.N_e$= 1.2, 1.3, 2.3, 2.4, 2.5. We apply different mincount filterings to each class.

Our first working grammar was built by excluding patterns reported in Table 2 and limiting the number of translations per source-side to 20. In brief, we have filtered out identical patterns (corresponding to rules with the same source and target pattern) and some monotonic non-terminal patterns (rule patterns in which non-terminals do not reorder from source to target). Identical patterns encompass a large number of rules and we have not been able to improve performance by using them in other translation tasks. Additionally, we have also applied mincount filtering to $N_{nt}.N_e$=1.3 and $N_{nt}.N_e$=2.4.

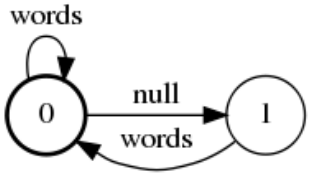

Figure 2: Consecutive Null filtering.

4.2 Deletion Rules

It has been experimentally found that statistical machine translation systems tend to benefit from allowing a small number of deletions. In other words, allowing some input words to be ignored (untranslated, or translated to NULL) can improve translation output. For this purpose, we add to our grammar one deletion rule for each source-language word, i.e. synchronous rules with the target side set to a special tag identifying the null word. In practice, this represents a huge increase in the search space as any number of consecutive words can be left untranslated. To control this undesired situtation, we would like to limit the number of consecutive deleted words to one. This is achieved by means of standard composition with a simple finite state transducer shown in Figure 2, where all words are allowed save for null word after having accepted one.

4.3 Hiero Shallow Model

A traditional Hiero grammar ($X \rightarrow \langle\gamma,\alpha\rangle$, $\gamma, \alpha \in (\{X\} \cup \mathbf{T})^+$, in addition to the glue rules) allows rule nesting only limited to the maximum word span. In contrast, the Hiero Shallow model allows hierarchical rules to be applied only once on top of phrase-based rules for the same given span. Stated more formally, for $s, t \in \mathbf{T}^+$ and $\gamma_s, \alpha_s \in (\{V\} \cup \mathbf{T})^+$ the Hiero Shallow grammar consists of three kind of rules:

$$X \rightarrow \langle\gamma_s,\alpha_s\rangle$$
$$X \rightarrow \langle V,V\rangle$$
$$V \rightarrow \langle s,t\rangle$$

Shifting from one grammar to another is as simple as rewriting X non-terminals in γ, α to V. It is easy to imagine the impact this has in terms of speed as it reduces drastically the size of the search space. In this sense, using Hiero Shallow Grammar can be seen as a filtering technique. Whether it has a negative impact on performance or not depends on each translation task: for instance, it was not useful for Chinese-to-English, as this task takes advantage of nested rules to find better reorderings encompassing a large number of words. On the other hand, a Spanish-to-English translation task is not expected to require big reorderings: thus, as a premise it is a good candidate for this kind of grammars. In effect, Table 3 shows that a hierarchical shallow grammar yields the same perfomance as full hierarchical translation.

Hiero Model	*dev2006*	*test2008*
Shallow	33.65/7.852	33.65/7.877
Full	33.63/7.849	33.66/7.880

Table 3: Performance of Hiero Full versus Hiero Shallow Grammars.

NT	*dev2006*	*test2008*
20	33.65/7.852	33.65/7.877
30	33.61/7.849	33.75/7.896
40	33.63/7.853	33.73/7.883

Table 4: Performance of G1 when varying the filter by number of translations, NT.

4.4 Filtering by Number of Translations

Filtering rules by a fixed number of translations per source-side (NT) allows faster decoding with the same perfomance. As stated before, the previous experiments for this task used a convenient baseline filtering of 20 translations. In our experience, this has been a good threshold for the NIST 2008 Arabic-to-English and Chinese-to-English translation tasks (Iglesias et al., 2009b; Iglesias et al., 2009a). In Table 4 we compare performance of our shallow grammar with different filterings, i.e. by 30 and 40 translations respectively. Interestingly, the grammar with 30 translations yields a slight improvement, but widening to 40 translations does not improve the translation system in performance.

4.5 Revisiting Patterns and Class Mincounts

In order to review the grammar design decisions taken in Section 4, and assess their impact in translation quality, we consider three competing grammars, i.e. $G1$, $G2$

	dev2006	*test2008*
$G1$	33.65/7.852	33.65/7.877
$G2$	33.47/7.838	33.65/7.877
$G3$	33.09/7.787	33.14/7.808

Table 5: Contrastive performance with three slightly different grammars.

and $G3$. $G1$ is the shallow grammar with $NT = 20$ already used (baseline). $G2$ is a subset of $G1$ (3.65M rules) with mincount filtering of 5 applied to $N_{nt}.N_e = 2.3$ and $N_{nt}.N_e = 2.5$. With this smaller grammar (3.25M rules) we would like to evaluate if we can obtain the same performance. $G3$ (4.42M rules) is a superset of $G1$ where the identical pattern $\langle X_1 w, X_1 w \rangle$ has been added. Table 5 shows translation performance with each of them. Decrease in performance for $G2$ is not surprising. These rules filtered out from $G2$ belong to reordered non-terminal rule patterns ($N_{nt}.N_e = 2.3$ and $N_{nt}.N_e = 2.5$) and some highly lexicalized monotonic non-terminal patterns from $N_{nt}.N_e = 2.5$, with three subsequence of words. More interesting is the comparison between $G1$ and $G3$, where we see that this extra identical rule pattern produces a degradation in performance.

5 Rescoring and Final Results

After translation with optimized feature weights, we carry out the two following rescoring steps to the output lattice.

- *Large-LM rescoring.* We build sentence-specific zero-cutoff stupid-backoff (Brants et al., 2007) 5-gram language models, estimated using ∼4.7B words of English newswire text, and apply them to rescore word lattice generated by HiFST.
- *Minimum Bayes Risk (MBR).* We rescore the first 1000-best hypotheses with MBR, taking the negative sentence level BLEU score as the loss function (Kumar and Byrne, 2004).

Table 6 shows results for our best Hiero model so far (using $G1$ with $NT = 30$) and subsequent rescoring steps. Gains from large language models are more modest than MBR, possibly due to the domain discrepancy between the EuroParl and the additional newswire data. Table 7 contains

	dev2006	*test2008*
HiFST	33.61/7.849	33.75/7.896
+5gram	33.66 /7.902	33.90/7.954
+MBR	33.87 /7.901	34.24/7.962

Table 6: EuroParl Spanish-to-English translation results (lower-cased IBM BLEU / NIST) after MET and subsequent rescoring steps

examples extracted from *dev2006*. Scores are state-of-the-art, comparable to the top submissions in the $WMT08$ shared-task results (Callison-Burch et al., 2008).

6 Conclusions

HiFST is a hierarchical phrase-based translation system that builds target word lattices. Based on well known standard WFST operations (such as union, concatenation and composition, among others), it is easy to implement, e.g. with the OpenFST library. The compact representation of multiple translation hypotheses in lattice form requires less pruning and performs better hypotheses recombination (i.e. by standard determinization) than cube pruning decoders, yielding fewer search errors and reduced overall memory use over k-best lists. For the EuroParl Spanish-to-English translation task, we have shown that Hiero grammars perform similarly to Hiero Shallow grammars, which only allow one level of hierarchical rules and thus yield much faster decoding times. We also present results with language-model rescoring and MBR. The performance of our system is state-of-the-art for Spanish-to-English (Callison-Burch et al., 2008).

References

Allauzen, Cyril, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. OpenFst: A general and efficient weighted finite-state transducer library. In *Proceedings of CIAA*, pages 11–23.

Bangalore, Srinivas and Giuseppe Riccardi. 2001. A finite-state approach to machine translation. In *Proceedings of NAACL*.

Bender, Oliver, Evgeny Matusov, Stefan Hahn, Sasa Hasan, Shahram Khadivi, and Hermann Ney. 2007. The RWTH Arabic-to-English spoken language trans-

Spanish	*English*
Estoy de acuerdo con él en cuanto al papel central que debe conservar en el futuro la comisión como garante del interés general comunitario.	I agree with him about the central role that must be retained in the future the commission as guardian of the general interest of the community.
Por ello, creo que es muy importante que el presidente del eurogrupo -que nosotros hemos querido crear- conserve toda su función en esta materia.	I therefore believe that it is very important that the president of the eurogroup - which we have wanted to create - retains all its role in this area.
Creo por este motivo que el método del convenio es bueno y que en el futuro deberá utilizarse mucho más.	I therefore believe that the method of the convention is good and that in the future must be used much more.

Table 7: Examples from the EuroParl Spanish-to-English dev2006 set.

lation system. In *Proceedings of ASRU*, pages 396–401.

Blunsom, Phil, Trevor Cohn, and Miles Osborne. 2008. A discriminative latent variable model for statistical machine translation. In *Proceedings of ACL-HLT*, pages 200–208.

Brants, Thorsten, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proceedings of EMNLP-ACL*, pages 858–867.

Callison-Burch, Chris, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2008. Further meta-evaluation of machine translation. In *Proceedings of the ACL Workshop on Statistical Machine Translation*, pages 70–106.

Casacuberta, Francisco. 2001. Finite-state transducers for speech-input translation. In *Proceedings of ASRU*.

Chappelier, Jean-Cédric and Martin Rajman. 1998. A generalized CYK algorithm for parsing stochastic CFG. In *Proceedings of TAPD*, pages 133–137.

Chappelier, Jean-Cédric, Martin Rajman, Ramón Aragüés, and Antoine Rozenknop. 1999. Lattice parsing for speech recognition. In *Proceedings of TALN*, pages 95–104.

Chiang, David. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Dyer, Christopher, Smaranda Muresan, and Philip Resnik. 2008. Generalizing word lattice translation. In *Proceedings of ACL-HLT*, pages 1012–1020.

Graehl, Jonathan, Kevin Knight, and Jonathan May. 2008. Training tree transducers. *Computational Linguistics*, 34(3):391–427.

Huang, Liang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Proceedings of ACL*, pages 144–151.

Iglesias, Gonzalo, Adrià de Gispert, Eduardo R. Banga, and William Byrne. 2009a. Hierarchical phrase-based translation with weighted finite state transducers. In *Proceedings of NAACL*, pages 433–441.

Iglesias, Gonzalo, Adrià de Gispert, Eduardo R. Banga, and William Byrne. 2009b. Rule filtering by pattern for efficient hierarchical translation. In *Proceedings of EACL*, pages 380–388.

Kneser, R. and H. Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of ICASSP*, volume 1, pages 181–184.

Koehn, Philipp, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL*, pages 48–54.

Kumar, Shankar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of HLT-NAACL*, pages 169–176.

Kumar, Shankar and William Byrne. 2005. Local phrase reordering models for statistical machine translation. In *Proceedings of HLT-EMNLP*, pages 161–168.

Li, Zhifei and Sanjeev Khudanpur. 2008. A scalable decoder for parsing-based machine translation with equivalent language model state maintenance. In *Proceedings of the ACL-HLT Second Workshop on Syntax and Structure in Statistical Translation*, pages 10–18.

Lopez, Adam. 2008. Tera-scale translation models via pattern matching. In *Proceedings of COLING*, pages 505–512.

Marton, Yuval and Philip Resnik. 2008. Soft syntactic constraints for hierarchical phrased-based translation. In *Proceedings of ACL-HLT*, pages 1003–1011.

Mathias, Lambert and William Byrne. 2006. Statistical phrase-based speech translation. In *Proceedings of ICASSP*.

Mohri, Mehryar, Fernando Pereira, and Michael Riley. 2000. The design principles of a weighted finite-state transducer library. *Theoretical Computer Science*, 231:17–32.

Och, Franz J. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL*, pages 160–167.

Rosti, Antti-Veikko, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie Dorr. 2007. Combining outputs from multiple machine translation systems. In *Proceedings of HLT-NAACL*, pages 228–235.

Shen, Libin, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL-HLT*, pages 577–585.

Sim, Khe Chai, William Byrne, Mark Gales, Hichem Sahbi, and Phil Woodland. 2007. Consensus network decoding for statistical machine translation system combination. In *Proceedings of ICASSP*, volume 4, pages 105–108.

Venugopal, Ashish, Andreas Zollmann, and Vogel Stephan. 2007. An efficient two-pass approach to synchronous-CFG driven statistical MT. In *Proceedings of HLT-NAACL*, pages 500–507.

Zhang, Hao and Daniel Gildea. 2006. Synchronous binarization for machine translation. In *Proceedings of HLT-NAACL*, pages 256–263.

Zollmann, Andreas and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of NAACL Workshop on Statistical Machine Translation*, pages 138–141.

Zollmann, Andreas, Ashish Venugopal, Franz Och, and Jay Ponte. 2008. A systematic comparison of phrase-based, hierarchical and syntax-augmented statistical MT. In *Proceedings of COLING*, pages 1145–1152.