FALA 2010
VI Jornadas en Tecnología del Habla and II Iberian SLTech Workshop

Hierarchical Phrase-based Translation with Weighted Finite-State Transducers

Gonzalo Iglesias[⋆‡]
Advisors: *Adrià de Gispert*[⋆] and *Eduardo R. Banga*[‡]
[⋆]Department of Engineering, University of Cambridge, CB2 1PZ, U.K.
[‡]Dept. of Signal Theory and Communications, University of Vigo, Spain
gi212@eng.cam.ac.uk

Abstract

This thesis [1] on Statistical Machine Translation (SMT) has shown it is possible to combine succesfully hierarchical decoding with finite-state technology. On the other side, refinements on hierarchical grammars lead to faster translations with similar performance. The combined strategy has worked very well on several language pairs, including Spanish-English translation tasks, leading in many cases to systems that are search-error free, with state-of-the-art performance.

Index Terms: statistical machine translation, hierarchical decoding, finite-state transducers.

1. Introduction

Hierarchical phrase-based translation (*Hiero*) is one of the dominant current approaches to SMT [2]. *Hiero* systems not only incorporate many of the strengths of phrase-based systems, but also allow flexible word reordering based on a hierarchical grammar, which is a specific instance of a synchronous context-free grammar consisting of a set of rules $X \rightarrow \langle\gamma,\alpha\rangle$ and two special 'glue' rules to allow monotonic concatenation [3].

Hierarchical systems apply these rules guided by a context-free parser. Although the underlying idea is that both source and target languages should have very similar 'syntactic' trees, the use of abstract non-terminal symbols rather than more linguistically motivated ones endows this strategy with several advantages: importantly, it is possible to extract automatically the grammar in a similar way to phrase-based rules, but now also considering word gaps, and with several added constraints to ensure feasibility [4].

The cube-pruning decoder is a common decoding strategy used to handle these grammars. For these systems, a monolingual source parse is performed first; then, by traversing backpointers of the parse forest, the translation search space is built. During this construction the cube-pruning technique is used in each cell, in order to make the procedure tractable; but of course not only this comes at a cost in terms of speed. As search errors inevitably appear, there is also a risk of degrading performance.

In this context, this thesis [1] shows that it is possible to use compact efficient lattice representations of the translation hypotheses within hierarchical decoding that lead to search-error free translation strategies. Furthermore, we use weighted finite-state transducers (WFSTs) to represent these lattices, with the advantage of powerful and efficient operations such as determinization, minimization or composition over WFSTs [5]. The result is a decoder named *HiFST*. Combined with a special refinement to hierarchical grammars which we call *shallow grammars*, *HiFST* has performed very well on several translation tasks.

This paper is organized as follows. In Section 2 we describe *HiFST*. In Section 3 we explain shallow-N grammars. In order to briefly assess this strategy, in Section 4 we discuss results in Chinese-English, Arabic-English and Spanish-English translation tasks, after which we conclude.

2. Hierarchical Decoding with WFSTs

The first step of this translation system is based on a variant of the CYK algorithm closely related to CYK+ [6]. We keep backpointers and employ hypotheses recombination without discarding rules. The underlying model is 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, with 'glue' rules, $S \rightarrow \langle X,X\rangle$ and $S \rightarrow \langle S\ X,S\ X\rangle$. If a rule has probability p^r, it is transformed to a cost c^r; here we use the tropical semiring, so $c^r = -\log p^r$. N denotes a priori any non-terminal (S,X,V, etcetera), $N \in \mathbf{N}$. $\mathbf{T}$ denotes the terminals (words), and the grammar builds parse forests based on strings $\gamma, \alpha \in \{\mathbf{N} \cup \mathbf{T}\}^+$. 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.

In effect, the source language sentence is parsed using a context-free grammar with rules $N \rightarrow \gamma$. The generation of translations is a second step that follows parsing. For this second step, we describe a method to construct word lattices with all possible translations that can be produced by the hierarchical rules. Construction proceeds by traversing the CYK grid along the backpointers established in parsing. In each cell (N, x, y) in 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. These lattices also contain the translation scores on their arc weights.

The ultimate objective is the word lattice $\mathcal{L}(S, 1, J)$, which corresponds to all the analyses that cover the source sentence s_1^J. Once this is built, we can apply a target language model to $\mathcal{L}(S, 1, J)$ to obtain the final target language translation lattice [7].

2.1. Lattice Construction Over the CYK Grid

In each cell (N, x, y), the set of rule indices used by the parser is denoted $R(N, x, y)$, *i.e.* for $r \in R(N, x, y)$, $N \rightarrow \langle\gamma^r,\alpha^r\rangle$ was used in at least one derivation involving that cell.

For each rule $R^r, r \in R(N, x, y)$, we build a lattice $\mathcal{L}(N, x, y, r)$. This lattice is derived from the target side of

⋆‡The work in this thesis has been done while Gonzalo Iglesias was still working for the Department of Signal Theory and Communications, at the University of Vigo.

the rule α^r by concatenating lattices corresponding to the elements of $\alpha^r = \alpha^r_1 ... \alpha^r_{|\alpha^r|}$. If an α^r_i is a terminal, creating its lattice is straightforward. If α^r_i is a non-terminal, it refers to a cell (N', x', y') lower in the grid identified by the backpointer $BP(N, x, y, r, i)$; in this case, the lattice used is $\mathcal{L}(N', x', y')$. Taken together,

$$\mathcal{L}(N, x, y, r) = \bigotimes_{i=1..|\alpha^r|} \mathcal{L}(N, x, y, r, i) \quad (1)$$

$$\mathcal{L}(N, x, y, r, i) = \begin{cases} \mathcal{A}(\alpha_i) & \text{if } \alpha_i \in \mathbf{T} \\ \mathcal{L}(N', x', y') & \text{else} \end{cases} \quad (2)$$

where $\mathcal{A}(t)$, $t \in \mathbf{T}$ returns a single-arc acceptor that accepts only the symbol t. The lattice $\mathcal{L}(N, x, y)$ is then built as the union of lattices corresponding to the rules in $R(N, x, y)$:

$$\mathcal{L}(N, x, y) = \bigoplus_{r \in R(N,x,y)} \mathcal{L}(N, x, y, r) \otimes c^r \quad (3)$$

This slight abuse of notation indicates that the cost c^r is applied at the path level to each lattice $\mathcal{L}(N, x, y, r)$; the cost can be added to the exit states, for example. This could as well be done at Equation 1.

2.2. Avoiding Pruning in Search

Equation 2 leads to the recursive construction of lattices in upper-levels of the grid through the union and concatenation of lattices from lower levels. If Equations 1 and 3 are actually carried out over fully expanded word lattices, the memory required by the upper lattices will increase exponentially.

To avoid this, we use special arcs that serve as pointers to the low-level lattices. This effectively builds a skeleton of the desired lattice and delays the creation of the final word lattice until a single replacement operation is carried out in the top cell $(S, 1, J)$. To make this exact, we define a function $g(N, x, y)$ that returns a unique tag for each lattice in each cell, and use it to redefine Equation 2. With the backpointer $(N', x', y') = BP(N, x, y, r, i)$, these special arcs are introduced as:

$$\mathcal{L}(N, x, y, r, i) = \begin{cases} \mathcal{A}(\alpha_i) & \text{if } \alpha_i \in \mathbf{T} \\ \mathcal{A}(g(N', x', y')) & \text{else} \end{cases} \quad (4)$$

The resulting lattices $\mathcal{L}(N, x, y)$ are a mix of target language words and lattice pointers. However, each still represents the entire search space of all translation hypotheses covering the span.

At the upper-most cell, the lattice $\mathcal{L}(S, 1, J)$ contains pointers to lower-level lattices. A single FST replace operation [5] recursively substitutes all pointers by their lower-level lattices until no pointers are left, thus producing the complete target word lattice for the whole source sentence. The use of the lattice pointer arc was inspired by the 'lazy evaluation' techniques developed by Mohri et al. [8]. Its implementation uses the infrastructure provided by the OpenFST libraries for delayed composition, etc.

Importantly, operations on these cell lattices — such as lossless size reduction via determinization and minimization — can still be performed. Owing to the existence of multiple hierarchical rules which share the same low-level dependencies, these operations can greatly reduce the size of the skeleton lattice; Figure 1 shows the effect on the translation example. This strategy is a key aspect to avoid pruning in search as much as possible. As stated, size reductions can be significant. However, not all redundancy is removed, since duplicate paths may arise through the concatenation and union of sublattices with different spans.

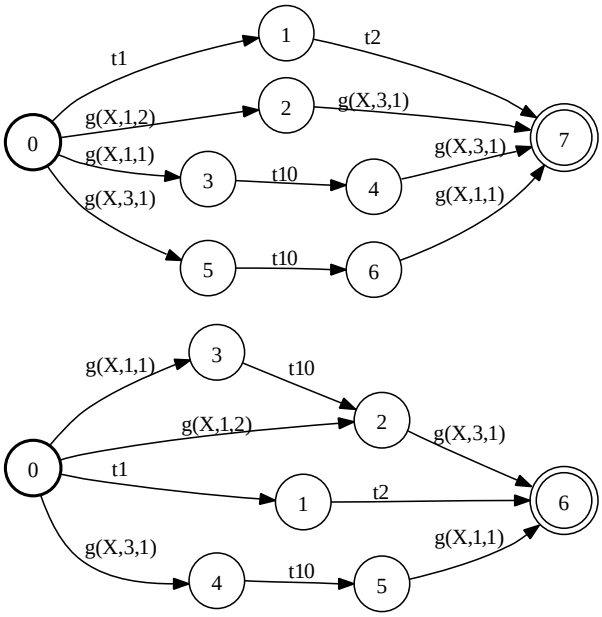

Figure 1: Delayed translation WFST with derivations from Figure 1 and Figure 2 before [t] and after minimization [b].

One interesting issue is where to use and where *not* to use pointer arcs. Several WFST operations are quite efficient due to the use of epsilon arcs. Unfortunately, combining carelessly these operations introduces an excessive number of epsilon arcs that very easily lead to intractable lattices. In many cases, removing epsilons is enough. But the expansion is a single operation that recursively traverses all the arcs substituting pointers to lower lattices by adding at least two epsilons per substitution. So, the issue is not only about making the lattice construction fast, but delivering a tractable skeleton for posterior steps. We decide which cell lattice will be replaced by a single arc depending on the non-terminal this cell is associated to. As a rule of thumb, the S cell lattices should never be replaced by pointer arcs, as they are used recursively many times for each translation hypothesis. A lattice construction doing so would return a minimal FST of two states binded by one single pointer arc, from which the complete search space lattice (possibly with millions of derivations) must be created, including at least twice as many epsilons as glue rules used within each derivation.

3. Shallow-N Grammars

Hierarchical grammars have shown good performance on translation tasks that demand lots of word reordering, such as Chinese-to-English. But for translation tasks between closer languages – e.g. Spanish-English – a search space defined by a hierarchical grammar leads to overgeneration, i.e. nonsensical translation hypotheses allowed by this powerful word reordering. Whereas hierarchical grammar allows any number of nestings through X rules up to a given word span threshold (typically 10 words), shallow-N grammars limit the size of the search space by defining a parameter N that controls directly the number of rule nestings allowed. More formally, a shallow-

N translation grammar can be defined as:

1. the usual non-terminal S
2. a set of non-terminals $\{X^0, \ldots, X^N\}$
3. two glue rules: $S \rightarrow \langle X^N, X^N \rangle$ and $S \rightarrow \langle S\ X^N, S\ X^N \rangle$
4. hierarchical translation rules for levels $n = 1, \ldots, N$:
   R: $X^n \rightarrow \langle \gamma, \alpha, \sim \rangle$, $\gamma, \alpha \in \{\{X^{n-1}\} \cup \mathbf{T}\}^+$
   with the requirement that α and γ contain at least one X^{n-1}
5. translation rules which generate lexical phrases:
   R: $X^0 \rightarrow \langle \gamma, \alpha \rangle$, $\gamma, \alpha \in \mathbf{T}^+$

Table 1 illustrates the shallow grammars for $N = 1, 2, 3$. As is clear, with larger N the expressive power of the grammar grows closer to that of full Hiero.

grammar	rules included
S-1	$\text{S} \rightarrow \langle X^1, X^1 \rangle \quad \text{S} \rightarrow \langle \text{S}\ X^1, \text{S}\ X^1 \rangle$ $X^0 \rightarrow \langle \gamma, \alpha \rangle$, $\gamma, \alpha \in \mathbf{T}^+$ $X^1 \rightarrow \langle \gamma, \alpha, \sim \rangle$, $\gamma, \alpha \in \{\{X^0\} \cup \mathbf{T}\}^+$
S-2	$\text{S} \rightarrow \langle X^2, X^2 \rangle \quad \text{S} \rightarrow \langle \text{S}\ X^2, \text{S}\ X^2 \rangle$ $X^0 \rightarrow \langle \gamma, \alpha \rangle$, $\gamma, \alpha \in \mathbf{T}^+$ $X^1 \rightarrow \langle \gamma, \alpha, \sim \rangle$, $\gamma, \alpha \in \{\{X^0\} \cup \mathbf{T}\}^+$ $X^2 \rightarrow \langle \gamma, \alpha, \sim \rangle$, $\gamma, \alpha \in \{\{X^1\} \cup \mathbf{T}\}^+$
S-3	$\text{S} \rightarrow \langle X^3, X^3 \rangle \quad \text{S} \rightarrow \langle \text{S}\ X^3, \text{S}\ X^3 \rangle$ $X^0 \rightarrow \langle \gamma, \alpha \rangle$, $\gamma, \alpha \in \mathbf{T}^+$ $X^1 \rightarrow \langle \gamma, \alpha, \sim \rangle$, $\gamma, \alpha \in \{\{X^0\} \cup \mathbf{T}\}^+$ $X^2 \rightarrow \langle \gamma, \alpha, \sim \rangle$, $\gamma, \alpha \in \{\{X^1\} \cup \mathbf{T}\}^+$ $X^3 \rightarrow \langle \gamma, \alpha, \sim \rangle$, $\gamma, \alpha \in \{\{X^2\} \cup \mathbf{T}\}^+$

Table 1: Rules contained in shallow-N grammars for $N = 1, 2, 3$.

Actually, shallow grammars are created by a trivial rewriting procedure of the full grammar. Consider the following example with a source sentence '$s_1\ s_2$' and a full grammar defined by these four rules:

R^1: $S \rightarrow \langle X, X \rangle$
R^2: $X \rightarrow \langle s_1\ s_2, t_2\ t_1 \rangle$
R^3: $X \rightarrow \langle s_1\ X, X\ t_1 \rangle$
R^4: $X \rightarrow \langle s_2, t_2 \rangle$

We can easily rewrite these rules according to a shallow-1 grammar:

R^1: $S \rightarrow \langle X^1, X^1 \rangle$
R^2: $X^0 \rightarrow \langle s_1\ s_2, t_2\ t_1 \rangle$
R^3: $X^1 \rightarrow \langle s_1\ X^0, X^0\ t_1 \rangle$
R^4: $X^0 \rightarrow \langle s_2, t_2 \rangle$
R^5: $X^1 \rightarrow \langle X^0, X^0 \rangle$

One interesting feature comes from the topology of shallow grammars: as they use several non-terminals, it is possible to set different constraints on each of them, e.g. enforce a minimum source word span. This is useful to speed up translation systems with shallow grammars allowing more than one nested rule [9].

4. Results

We have assessed the validity of our combined strategy throughout several translations tasks, such as Chinese-to-English, Arabic-to-English, and Spanish-English, amongst others. In first place, we have shown that *HiFST*, even when pruning-in-search is required, never makes more search errors than a hierarchical cube-pruning decoder, overall resulting in better performance. This is specially notable for rescoring steps, after which an improvement of more than 1 BLEU point for both Arabic-to-English and Chinese-to-English is achieved [9]. Whereas with a complex task such as Chinese-to-English a *full* hierarchical grammar is needed to achieve the best performance, a nesting of 2 or even 1 is enough for other closer language pairs. All these findings are discussed in detail throughout several conference papers [10, 11, 12] and a journal paper [9]. *HiFST* is also the core of the CUED system, which ranked first in the Arabic-to-English NIST 2009 Constrained Data Track[1].

In this section we will focus on some relevant results of the Spanish-English translation tasks.

The training was performed using lower-cased data. Word alignments were generated using GIZA++ [13] over a stemmed 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 [14]. Hierarchical rules with up to two non-contiguous non-terminals in the source side are then extracted applying the usual restrictions [4]. The Europarl language model is a Kneser-Ney [15] smoothed default cutoff 4-gram back-off language model estimated over the concatenation of the Europarl and News language model training data.

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

- *Large-LM rescoring (5g)*. We build sentence-specific zero-cutoff stupid-backoff [16] 5-gram language models.
- *Minimum Bayes Risk (MBR)*. We rescore the first 1000-best hypotheses with MBR [17], or the lattice with Lattice MBR (LMBR) [18], taking the negative sentence level BLEU score as the loss function.

4.1. Experiments on the Shared task of WMT08

In this subsection we present experiments for Spanish-to-English on the shared task of the ACL 2008 Workshop on Statistical Machine Translation [12].

As we had already discovered for the Arabic-to-English task [10], we found that the shallow-1 grammar already had the same performance as hierarchical full system at much greater speed, as pruning during search is avoided entirely. Table 2 shows results for our shallow-1 model 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.

Scores are comparable to the top submissions in the $WMT08$ shared-task results [19].

	dev2006	*test2008*
HiFST(S-1)	33.6/7.85	33.8/7.90
+5g	33.7 /7.90	33.9/7.95
+5g+MBR	33.9 /7.90	34.2/7.96

Table 2: EuroParl Spanish-to-English translation results (lower-cased IBM BLEU / NIST) after MET and subsequent rescoring steps

[1] See http://www.itl.nist.gov/iad/mig/tests/mt/2009/ResultsRelease for full MT09 results.

Task	System	*nwtest08*	*nwtest09*	*nwtest10*
SP → EN	HiFST(S-1)	24.6	26.0	29.1
	+5g+LMBR	25.4	27.0	**30.5**
EN → SP	HiFST (S-1)	23.9	24.5	28.0
	+5g+LMBR	24.7	25.5	**29.1**

Table 3: Translation Results for the Spanish-English (SP-EN) language pair, shown in lowercase IBM BLEU. Bold results correspond to submitted systems.

4.2. Shared task of WMT10

We have participated in the ACL 2010 Workshop of Machine Translation [20] on several translation tasks [21]. Table 3 shows the excellent results for both Spanish-to-English and English-to-Spanish tasks. In both directions, using *HiFST* with shallow-1 grammars allows a search-error free decoding. In turn, this allows rescoring steps to increase the performance in more than 1 BLEU point for both directions.

5. Conclusions

In the context of hierarchical translation, this thesis has proposed a novel translation system that uses WFSTs within hierarchical decoding, capable of compact and efficient representations of the translation search space. A refinement to the hierarchical grammars, which we call shallow-N grammars, has also been introduced. This refinement allows a simple tuning to the word-reordering requirements of each particular translation task, thus avoiding overgeneration. Taken both strategies together, it is possible to build state-of-the-art translation systems between close languages – such as Spanish, English, French or even Arabic – without pruning in search, leading to faster decoding times, search-error free translation lattices and improved rescoring performance. This thesis is available for download at `http://www.eng.cam.ac.uk/~gi212/thesis.pdf`.

6. Acknowledgements

This work has been supported by Spanish Government research grant BES-2007-15956, project AVIVAVOZ (TEC2006-13694-C03-03) and project BUCEADOR (TEC2009-14094-C04-04). Also supported in part by the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022.

7. References

[1] G. Iglesias, "Hierarchical phrase-based translation with weighted finite-state transducers," Ph.D. dissertation, University of Vigo, Spain, 2010.

[2] A. Zollmann, A. Venugopal, F. Och, and J. Ponte, "A systematic comparison of phrase-based, hierarchical and syntax-augmented statistical MT," in *Proceedings of COLING*, 2008, pp. 1145–1152.

[3] D. Chiang, "A hierarchical phrase-based model for statistical machine translation," in *Proceedings of ACL*, 2005, pp. 263–270.

[4] ——, "Hierarchical phrase-based translation," *Computational Linguistics*, vol. 33, no. 2, pp. 201–228, 2007.

[5] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "OpenFst: A general and efficient weighted finite-state transducer library," in *Proceedings of CIAA*, 2007, pp. 11–23.

[6] J.-C. Chappelier and M. Rajman, "A generalized CYK algorithm for parsing stochastic CFG," in *Proceedings of TAPD*, 1998, pp. 133–137.

[7] C. Allauzen, M. Mohri, and B. Roark, "Generalized algorithms for constructing statistical language models," in *Proceedings of ACL*, 2003, pp. 557–564.

[8] M. Mohri, F. Pereira, and M. Riley, "The design principles of a weighted finite-state transducer library," *Theoretical Computer Science*, vol. 231, pp. 17–32, 2000.

[9] A. de Gispert, G. Iglesias, G. Blackwood, E. R. Banga, and W. Byrne, "Hierarchical phrase-based translation with weighted finite-state transducers and shallow-n grammars," *Computational Linguistics*, vol. 36, no. 3, pp. 505–533, 2010.

[10] G. Iglesias, A. de Gispert, E. R. Banga, and W. Byrne, "Rule filtering by pattern for efficient hierarchical translation," in *Proceedings of EACL*, 2009, pp. 380–388.

[11] ——, "Hierarchical phrase-based translation with weighted finite state transducers," in *Proceedings of NAACL-HLT*, 2009, pp. 433–441.

[12] ——, "The HiFST system for the europarl Spanish-to-English task," in *Proceedings of SEPLN*, 2009, pp. 207–214.

[13] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[14] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of NAACL-HLT*, 2003.

[15] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proceedings of ICASSP*, vol. 1, 1995, pp. 181–184.

[16] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean, "Large language models in machine translation," in *Proceedings of EMNLP-ACL*, 2007, pp. 858–867.

[17] S. Kumar and W. Byrne, "Minimum Bayes-risk decoding for statistical machine translation," in *Proceedings of NAACL-HLT*, 2004, pp. 169–176.

[18] R. Tromble, S. Kumar, F. J. Och, and W. Macherey, "Lattice Minimum Bayes-Risk decoding for statistical machine translation," in *Proceedings of EMNLP*, 2008, pp. 620–629.

[19] C. Callison-Burch, C. Fordyce, P. Koehn, C. Monz, and J. Schroeder, "Further meta-evaluation of machine translation," in *Proceedings of WMT*, 2008, pp. 70–106.

[20] C. Callison-Burch, P. Koehn, C. Monz, K. Peterson, M. Przybocki, and O. Zaidan, "Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation," in *Proceedings of WMT*, 2010, pp. 17–53.

[21] J. Pino, G. Iglesias, A. de Gispert, G. Blackwood, J. Brunning, and W. Byrne, "The CUED HiFST System for the WMT10 Translation Shared Task," in *Proceedings of WMT*, 2010, pp. 155–160.