Flexible and Efficient Hypergraph Interactions for Joint Hierarchical and Forest-to-String Decoding*

Martin Čmejrek†‡
†IBM Prague Research Lab
V Parku 2294/4
Prague, Czech Republic, 148 00
martin.cmejrek@us.ibm.com

Haitao Mi‡ and **Bowen Zhou**‡
‡IBM T. J. Watson Research Center
1101 Kitchawan Rd
Yorktown Heights, NY 10598
{hmi,zhou}@us.ibm.com

Abstract

Machine translation benefits from system combination. We propose *flexible interaction of hypergraphs* as a novel technique combining different translation models within one decoder. We introduce features controlling the interactions between the two systems and explore three interaction schemes of hiero and forest-to-string models—specification, generalization, and interchange. The experiments are carried out on large training data with strong baselines utilizing rich sets of dense and sparse features. All three schemes significantly improve results of any single system on four testsets. We find that specification—a more constrained scheme that almost entirely uses forest-to-string rules, but optionally uses hiero rules for shorter spans—comes out as the strongest, yielding improvement up to 0.9 (T -B)/2 points. We also provide a detailed experimental and qualitative analysis of the results.

1 Introduction

Recent years have witnessed the success of various statistical machine translation (SMT) models using different levels of linguistic knowledge–phrase (Koehn et al., 2003), hiero (Chiang, 2005), and syntax-based (Liu et al., 2006; Galley et al., 2006). System combination became a promising way of building up synergy from different SMT systems and their specific merits.

Numerous efforts that have been proposed in this field recently can be broadly divided into two categories: *Offline system combination* (Rosti et al., 2007; He et al., 2008; Watanabe and Sumita, 2011; Denero et al., 2010) aims at producing consensus translations from the outputs of multiple individual systems. Those outputs usually contain k-best lists of translations, which only explore a small portion of the entire search space of each system. This issue is well addressed in *joint decoding* (Liu et al., 2009), or *online system combination*, showing comparable improvements to the offline combination methods. Rather than finding consensus translations from the outputs of individual systems, joint decoding works with different grammars at the decoding time. Although limited to individual systems sharing the same search paradigm (e.g. *left-to-right* or *bottom-up*), joint decoding offers many potential advatages: search through a larger space, better efficiency, features designed once for all subsystems, potential cross-system features, online sharing of partial hypotheses, and many others.

Different approaches have different strengths in general–hiero rules are believed to provide reliable lexical coverage, while tree-to-string rules are good at non-local reorderings. Different contexts present different challenges–noun phrases usually follow the adjacency principle, while verb phrases require more challenging reorderings. In this work, we study different schemes of interaction between translation models, reflecting their specific strengths at different (syntactic) contexts. We make five new contributions:

> **First**, we propose a framework for joint decoding by means of flexible combination of translation hypergraphs, allowing for detailed con-

*M. Č and H. M. contributed equally to this work.

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 545–555,
Seattle, Washington, USA, 18-21 October 2013.

trol of interactions between the different systems using soft constraints (Section 3).

Second, we study three interaction schemes– special cases of joint decoding: *generalization*, *specification*, and *interchange* (Section 3.3).

Third, instead of using a tree-to-string system, we use a much stronger forest-to-string system with fuzzy match of nonterminal categories (Section 2.1).

Fourth, we train strong systems on a large-scale data set, and test all methods on four test sets. Experimental results (Section 6) show that our new approach brings improvement of up to 0.9 points in terms of (T − B)/2 over the best single system.

Fifth, we conduct a comprehensive experimental analysis, and find that joint decoding actually prefers tree-to-string rules in both shorter and longer spans. (Section 6.3).

The paper is organized as follows: We briefly review the individual models in Section 2, describe the method of joint decoding using three alternative interaction schemes in Section 3, describe the features controlling the interactions and fuzzy match in Section 4, review the related work in Section 5, and finally, describe our experiments and give detailed discussion of the results in Section 6.

2 Individual Models

Our individual models are two state-of-the-art systems: a hiero model (Chiang, 2005), and a forest-to-string model (Mi et al., 2008; Mi and Huang, 2008).

We will use the following example from Chinese to English to explain both individual and joint decoding algorithms throughout this paper.

tǎolùn	hùi	zěnmeyàng
discussion/NN	will/VV	how/VV
discuss/VV	meeting/NN	

There are several possible meanings based on the different POS tagging sequences:

1: NN VV VV: *How is the discussion going?*
2: VV NN VV: *Discuss about the meeting.*
3: NN NN VV: *How was the discussion meeting?*
4: VV VV VV: *Discuss what will happen.*

id	rule
r_1	VV(tǎolùn) → discuss
r_2	NP(tǎolùn) → the discussion
r_3	NP(hùi) → the meeting
r_4	VP(zěnmeyàng) → how
r'_4	VP(zěnmeyàng) → about
r_5	IP(x_1:NP x_2:VP) → x_2 x_1
r_6	IP(x_1:VV x_2:IP) → x_1 x_2
r_7	IP(x_1:NP VP(VV(hùi) x_2:VP)) → x_2 is x_1 going
r_{11}	X(x_1:X zěnmeyàng) → how was x_1
r_{12}	X(zěnmeyàng) → what
r_{13}	X(tǎolùn hùi) → the discussion meeting
r_{14}	X(hùi x_1:X) → x_1 will happen
r_{15}	S(x_1:S x_2:X) → x_1 x_2

Table 1: Translation rules. Tree-to-string (r_1–r_7), hiero (r_{11}–r_{14}), vanilla glue (r_{15}).

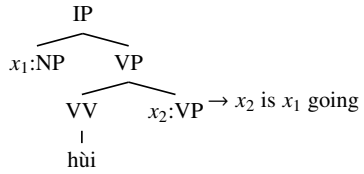


Figure 1: Tree-to-string rule r_7.

Table 1 shows translation rules that can generate all four translations. We will use those rules in the following sections.

2.1 Forest-to-string

Forest-to-string translation (Mi et al., 2008) is a linguistic syntax-based system, which significantly improves the translation quality of the tree-to-string model (Liu et al., 2006; Huang et al., 2006) by using a packed parse forest as the input instead of a single parse tree.

Figure 1 shows a tree-to-string **translation rule** (Huang et al., 2006), which is a tuple $\langle lhs(r), rhs(r), \psi(r)\rangle$, where $lhs(r)$ is the source-side tree fragment, whose internal nodes are labeled by nonterminal symbols (like NP and VP), and whose frontier nodes are labeled by source-language words (like "hùi") or variables from a set $\mathcal{X} = \{x_1, x_2, \ldots\}$; $rhs(r)$ is the target-side string expressed in target-language words (like "going") and variables; and $\psi(r)$ is a mapping from $\mathcal{X}$ to nonterminals. Each

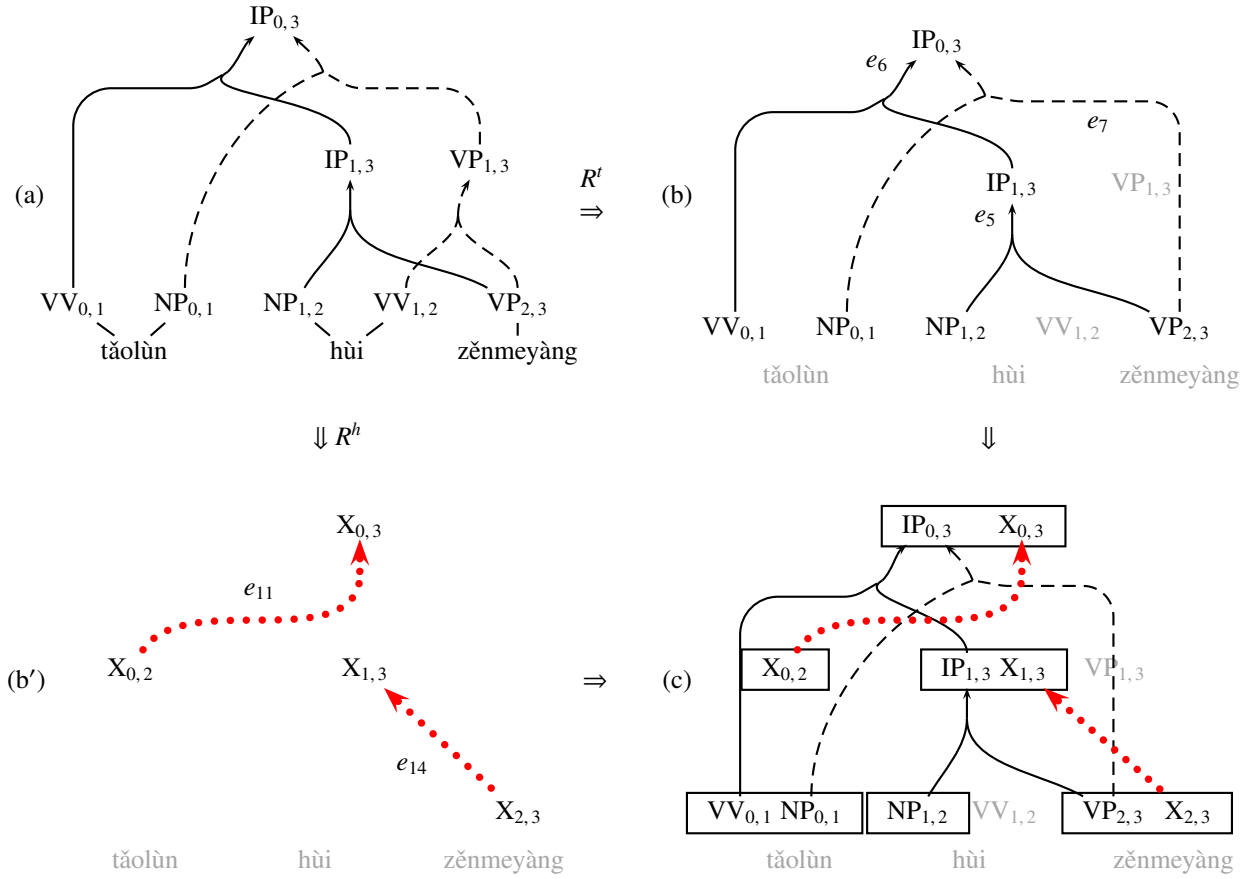


Figure 2: Parse and translation hypergraphs. (a) The parse forest of the example sentence. Solid hyperedges denote the 1-best parse. (b) The corresponding translation forest F^t after applying the tree-to-string translation rule set R^t. Target lexical content is not shown. Each translation hyperedge (e.g. e_7) has the same index as the corresponding rule (r_7). Gray nodes (e.g. $\text{VP}_{1,3}$) became inaccessible due to the insufficient rule coverage. (b′) The translation forest F^h after applying the hierarchical rule set R^h to the input sentence. (c) The combined translation forest H^m obtained by superimposing b and b′. The nodes within each solid box share the same span. See Figure 3 for an example of the internal structure of a box. The forest-to-string system can produce the translation 1 (dashed derivation: r_2, r_4 and r_7) and 2 (solid derivation: r_1, r_3, r'_4, r_5, and r_6). Hierarchical rules generate the translation 3 (r_{11} and r_{13}). The translation 4 is available by using joint decoding at $\text{X}_{1,3} \rightarrow \text{IP}_{1,3}$ with the derivation: r_1, r_6, r_{12}, and r_{14}.

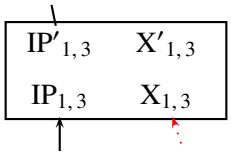
variable $x_i \in \mathcal{X}$ occurs *exactly once* in $lhs(r)$ and *exactly once* in $rhs(r)$. Take the rule r_7 in Figure 1 for example, we have:

$$lhs(r_7) = \text{IP}(x_1\text{:NP VP(VV(hùi) } x_2\text{:VP)}),$$
$$rhs(r_7) = x_2 \text{ is } x_1 \text{ going},$$
$$\psi(r_7) = \{x_1 \mapsto \text{NP}, x_2 \mapsto \text{VP}\}.$$

Typically, a forest-to-string system performs translation in two steps (shown in Figure 2): parsing and decoding. In the parsing step, we convert the source language input into a *parse forest* (a). In the decoding step, we first convert the parse forest into a *translation forest* F^t in (b) by using the fast pattern-matching technique (Zhang et al., 2009). For example, we pattern-match the rule r_7 rooted at $\text{IP}_{0,3}$, in such a way that x_1 spans $\text{NP}_{0,1}$ and x_2 spans $\text{VP}_{2,3}$, and add a translation hyperedge e_7 in (b). Then the decoder searches for the best derivation on the translation forest and outputs the target string.

2.2 Hiero

Hiero (hierarchical phrase-based) model (Chiang, 2005) acquires rules of synchronous context-free grammars (SCFGs) from word-aligned parallel data, and uses plain sequences of words as the input, without any syntactic information.



scheme	interaction edges in supernode
Generalization	IP′1, 3 X′1, 3 / IP1, 3 X1, 3
Specification	IP′1, 3 X′1, 3 / IP1, 3 X1, 3
Interchange	IP′1, 3 X′1, 3 / IP1, 3 X1, 3

Figure 3: Three interaction schemes for joint decoding. Details of the interaction supernode for span $(1, 3)$ shown in Figure 2 (c). Soft constraints control the transitions.

SCFG can be formalized as a set of tuples $\langle lhs(r), rhs(r), \phi(r)\rangle$, where $lhs(r)$ is the source-side one-level CFG, whose root is X or S, and whose frontier nodes are labeled by source-language words (like "hùi") or variables from a set $\mathcal{X} = \{x_1, x_2, \ldots\}$; $rhs(r)$ is the target-side string expressed in target-language words (like "going") and variables; and $\phi(r)$ is a mapping from $\mathcal{X}$ to nonterminals. Table 1 shows examples of hiero rules r_{11}–r_{15}.

Although different on source side, hiero decoding can be formalized equally as forest-to-string decoding: First, pattern-match the input sentence into a *translation forest* F^h. For example, since the rule r_{11} matches "zěnmeyàng" such that x_1 spans the first two words, add a hyperedge e_{11} in Figure 2 (b′). Then search for the best derivation over the translation forest.

3 Joint Decoding

The goal of joint decoding is to let different MT models collaborate within the framework of a single decoder. This can be done by combining translation hypergraphs of the different models at the decoding time, so that online sharing of partial hypotheses overcomes weaknesses and boosts strengths of the systems combined.

As both forest-to-string and hiero produce translation forests that share the same *hypergraph* structure, we first formalize the hypergraph, then we introduce an algorithm to combine different hypergraphs, and finally we describe three joint decoding schemes over the merged hypergraph.

3.1 Hypergraphs

More formally, a hypergraph H is a pair $\langle V, E\rangle$, where V is the set of **nodes**, and E the set of **hyperedges**. For a given sentence $w_{1:l} = w_1 \ldots w_l$, each node $v \in V$ is in the form of $Y_{i,j}$, where Y is a nonterminal in the context-free grammar[1] and i, j, $0 \le i < j \le l$, are string positions in the sentence $w_{1:l}$, which denote the recognition of nonterminal Y spanning the substring from positions i through j (that is, $w_{i+1} \ldots w_j$). Each hyperedge $e \in E$ is a tuple $\langle tails(e), head(e), target(e)\rangle$, where $head(e) \in V$ is the *consequent node* in the deductive step, $tails(e) \in V^*$ is the list of *antecedent nodes*, and $target(e)$ is a list of $rhs(r)$ for rules r such that each rule r has the same $lhs(r)$ pattern-matched at the node $head(e)$. For example, the hyperedge e_7 in Figure 2 (b) is

$$e_7 = \langle (\mathrm{NP}_{0,1}, \mathrm{VP}_{2,3}), \mathrm{IP}_{0,3}, (\mathrm{x}_2 \text{ is } x_1 \text{ going})\rangle,$$

where we can infer the mapping to be

$$\{x_1 \mapsto \mathrm{NP}_{0,1}, x_2 \mapsto \mathrm{VP}_{2,3}\}.$$

We also denote $BS(v)$ to be the set of **incoming hyperedges** of node v, which represent the different ways of deriving v. For example, $BS(\mathrm{IP}_{0,3})$ is a set of e_7 and e_6.

There is also a distinguished **root node** TOP in each hypergraph, denoting the goal item in translation, which is simply $\mathrm{TOP}_{0,l}$.

3.2 Combining Hypergraphs

We enable interaction between translation hypergraphs, such as hiero $F^h = \langle V^h, E^h\rangle$ and forest-to-string $F^t = \langle V^t, E^t\rangle$, on nodes covering the same span (e.g. $\mathrm{IP}_{1,3}$ and $\mathrm{X}_{1,3}$ in Figure 2 (c) grouped in a box). We call such groups *interaction supernodes* and show a detailed example of a supernode for span $(1, 3)$ in Figure 3.

The combination runs in four steps:

[1] In this paper, nonterminal labels X and S denote hiero derivations, other labels are tree-to-string labels.

1. For each node $v = Y_{i,j}, v \in V^h \cup V^t$, we create a new *interaction node* $v' = Y'_{i,j}$ with empty $BS(v')$. For example, we create two nodes, $\mathrm{IP}'_{1,3}$ and $\mathrm{X}'_{1,3}$, at the top of Figure 3.

2. For each hyperedge $e \in BS(v)$, $v \in V^t \cup V^h$, we replace each v in $tails(e)$ with v'. For example, e_7 becomes $\langle(\mathrm{NP}'_{0,1}, \mathrm{VP}'_{2,3}), \mathrm{IP}_{0,3}, (\mathrm{x}_2 \text{ is } x_1 \text{ going})\rangle$.

3. All the nodes and hyperedges form the merged hypergraph F^m, such as in Figure 2 (c).

4. Insert *interaction hyperedges* connecting nodes within each interaction supernode to make F^m connected again.

In the following subsection we present details of interactions and introduce three alternative schemes.

3.3 Three Schemes of Joint Decoding

Interaction hyperedges within each supernode allow the decoder either to stay within the same system (e.g. in hiero using $\mathrm{X}_{1,3} \rightarrow \mathrm{X}'_{1,3}$ in Figure 3), or to switch to the other (e.g. to forest-to-string using $\mathrm{X}_{1,3} \rightarrow \mathrm{IP}'_{1,3}$).

For example, translation 4 can be produced as follows: The source string "zěnmeyàng" is translated by the phrase rule r_{12}. The hiero hyperedge e_{14} combines it with the translation of "hùi", reaching the hiero node $\mathrm{X}_{1,3}$. Using the interaction edge $\mathrm{X}_{1,3} \rightarrow \mathrm{IP}'_{1,3}$ will switch into the tree-to-string model, so that the translation can be completed with the tree-to-string edge e_6 that connects it with a partial tree-to string translation of "tǎolùn" done by r_1.

In order to achieve more precise control over the interaction between tree-to-string and hiero derivations, we propose the following three basic interaction schemes: **generalization**, **specification**, **interchange**. The schemes control the interaction between hiero and tree-to-string models by means of soft constraints. Some schemes may even restrict certain types of transitions. The schemes are depicted in Figure 3 and their details are discussed in the following three subsections.

3.3.1 Specification

The *specification* decoding scheme reflects the intuition of using hiero rules to translate shorter spans and tree-to-string rules to reorder higher-level sentence structures. In other words, the scheme allows one-way switching from the hiero general nonterminal into the more *specific* nonterminal of a tree-to-string rule. Transitions in reverse directions are not allowed. This is achieved by inserting *specification interaction hyperedges e* leading from hiero nodes $\mathrm{X}_{i,j}$ or $\mathrm{S}_{i,j}$ into all tree-to-string interaction nodes $\mathrm{Y}'_{i,j}$ within the same supernode.

3.3.2 Generalization

In some translation domains, hiero outperforms tree-to-string systems, as was shown in experiments in Section 6. While local hiero or tree-to-string reorderings perform well, long distance reorderings proposed by tree-to-string may be too risky (e.g. due to parsing errors), so that monotone concatenation of long sequences[2] is the more reliable strategy. The *generalization* decoding scheme, complementary to the specification, is motivated by the idea of incorporating reliable tree-to-string translations for some sequences into a strong hiero translation system. This is achieved by inserting *generalization interaction hyperedges e* leading from tree-to-string nodes $\mathrm{Y}_{i,j}$ nodes into *general* hiero interaction nodes $\mathrm{X}'_{i,j}$ and $\mathrm{S}'_{i,j}$ within the same supernode.

3.3.3 Interchange

The *interchange* decoding scheme is a union of the two previous approaches. Any derivation can freely combine hiero and tree-to-string productions. Both *specification* and *generalization interaction hyperedges* are inserted leading from all hiero and tree-to-string nodes $\mathrm{X}_{i,j}$, $\mathrm{S}_{i,j}$, and $\mathrm{Y}_{i,j}$ into all interaction nodes $\mathrm{X}'_{i,j}$, $\mathrm{S}'_{i,j}$, and $\mathrm{Y}'_{i,j}$.

3.4 Fuzzy match

The translation rule set cannot usually cover all hyperedges in the parse forest, thus some nodes become inaccessible in the translation forest (e.g. $\mathrm{VP}_{1,3}$ in Figure 2). However, in the parse forest, as opposed to a 1-best tree, we can find other nodes spanning the same sequence $w_{i:j}$ (e.g. node $\mathrm{IP}_{1,3}$). In order to re-enable inaccessible nodes and to increase the variability of the translation forest, we allow reaching them from the other tree-to-string

[2]Monotone glue is the only possibility for very long spans exceeding the hiero *maxParse* treshold.

nodes within the same interaction node. This can be achieved by adding *fuzzy hyperedges* between every tree-to-string state $Y_{i,j}$ and a differently labeled tree-to-string interaction state $Z'_{i,j}$. For example, in the span (0,1), we have a fuzzy hyperedge $\mathrm{VV}_{0,1} \rightarrow \mathrm{NP}'_{0,1}$.

While interaction hyperedges combine different translation models, fuzzy hyperedges combine different derivations within the same (tree-to-string) model.

4 Interaction Features

Our baseline systems use the log-linear framework to estimate the probability $P(D)$ of a derivation D from features ϕ_i and their weights λ_i as $P(D) \propto \exp\left(\sum_i \lambda_i \phi_i\right)$. Similarly as Chiang et al. (2009), our systems use tens of dense (e.g. language models, translation probabilities) and thousands of sparse (e.g. lexical, fertility) features.

The features related to the joint decoding experiments are the costs for *specification*, *generalization*, *interchange*, and the *fuzzy match*. Let L^t be the set of the labels used by the source language parser and $L^h = \{\mathrm{S},\mathrm{X}\}$ be the labels used by hiero.

The **generalization feature**

$$\phi_{Y \rightarrow Z} = |\{e; e \in D, \exists i, j \; tails(e) = \{\mathrm{Y}_{i,j}\} \wedge head(e) = \mathrm{Z}'_{i,j}\}| \quad (1)$$

is the total number of generalization hyperedges in D going from tree-to-string states $Y \in L^t$ to hiero states $Z' \in L^h$.

The **specification feature**

$$\phi_{Z \rightarrow Y} = |\{e; e \in D, \exists i, j \; tails(e) = \{\mathrm{Z}_{i,j}\} \wedge head(e) = \mathrm{Y}'_{i,j}\}| \quad (2)$$

is the total number of specification hyperedges in D going from hiero states $Z \in L^h$ to tree-to-string states $Y' \in L^t$.

The **interchange feature** is implemented by enabling the generalization and specification features at the same time for both tuning and testing.

The **fuzzy match feature**

$$\phi_{U \rightarrow W} = |\{e; e \in D, \exists i, j \; tails(e) = \{\mathrm{U}_{i,j}\} \wedge head(e) = \mathrm{W}'_{i,j}\}| \quad (3)$$

is the total number of fuzzy match hyperedges in D going from tree-to-tree states $U \in L^t$ to tree-to-string states $W' \in L^t$. [3]

We use MIRA to obtain weights for the new features by tuning on the development set. The number of new parameters to tune can be estimated as $|L^h| \times |L^t|$ for generalization and specification, and $2 \times |L^h| \times |L^t|$ for interchange. For the fuzzy match of tree-to-string nonterminals we have $|L^t| \times |L^t|$ parameters organized as a sparse matrix, since we only consider combinations on nonterminal labels that cooccur in the data.[4]

5 Related Work

From the previous explorations of online translation model combination, we see the work of Liu et al. (2009) proposing an unconstrained combination of hiero and tree-to-string models as a special configuration of our framework, and we also replicate it.

Denero et al. (2010) combine translation models even with different search paradigms. Their approach is different, since their component systems do not interact at decoding time, instead, each of them provides its weighted translation forest first, the forests are then combined to infer a new combination model.

6 Experiment

In this section we describe the setup, present results, and analyze the experiments. Finally, we propose future directions of research.

[3] Here we allow $U = W$, which can be viewed in such a way that exact match is a special case of fuzzy match.

[4] We also carried out an alternative experiment with only three fuzzy match features estimated from the training data parse forest by Naïve Bayes by observing all spans in the training data, accumulating counts $C_s(U)$ and $C_s(U, W)$ of nonterminals (or pairs of nonterminals) heading the same span s. The first two features (one for each direction) are based on conditional probabilities:

$$\phi_{(U|W)} = -\log\left(\frac{\sum_{s \in spans} C_s(U, W)}{\sum_{s \in spans} C_s(W)}\right). \quad (4)$$

The third feature is based on joint probability:

$$\phi_{(U,W)} = -\log\left(\frac{\sum_{s \in spans} C_s(U, W)}{\sum_{s \in spans, A, B \in L^t} C_s(A, B)}\right). \quad (5)$$

The average performance drops by 0.1 (T -B)/2 points, compared to the interchange eperiment.

System		GALE-web		P1R6-web		MT08 news		MT08 web		Avg.
		B	(T-B)/2	B	(T-B)/2	B	(T-B)/2	B	(T-B)/2	(T-B)/2
Single	T2S	32.6	11.6	16.9	23.5	37.7	7.8	28.1	14.5	14.4
	Hiero	33.7	**10.2**	17.0	**23.1**	39.2	**6.3**	28.8	13.7	**13.3**
	F2S	34.0	10.3	17.3	23.2	39.6	**6.3**	29.2	**13.6**	13.4
Joint	Liu:09	34.1	9.7	17.0	23.0	38.8	6.7	29.0	13.2	13.2
	Gen.	34.4	9.7	17.8	22.6	40.0	6.1	29.6	13.1	12.9
	Spe.	35.1	**9.4**	18.1	**22.2**	40.2	**5.8**	29.6	**12.9**	**12.6**
	Int.	34.9	**9.4**	17.9	22.3	40.0	6.2	29.6	**12.9**	12.7

Table 2: All results of single and joint decoding systems.

6.1 Setup

The training corpus consists of 16 million sentence pairs available within the DARPA BOLT Chinese-English task. The corpus includes a mix of newswire, broadcast news, webblog and comes from various sources such as LDC, HK Law, HK Hansard and UN data. The Chinese text is segmented with a segmenter trained on CTB data using conditional random fields (CRF). Language models are trained on the English side of the parallel corpus, and on monolingual corpora, such as Gigaword (LDC2011T07) and Google News, altogether comprising around 10 billion words.

We use a modified version of the Berkeley parser (Petrov and Klein, 2007) to obtain a parse forest for each training sentence, then we prune it with the marginal probability-based inside-outside algorithm to contain only $3n$ CFG nodes, where n is the sentence length. Finally, we apply the forest-based GHKM algorithm (Mi and Huang, 2008; Galley et al., 2004) to extract tree-to-string translation rules from forest-string pairs.

In the decoding step, we prune the input hypergraphs to $10n$ nodes before we use fast pattern-matching (Zhang et al., 2009) to convert the parse forest into the translation forest.

We tune on 1275 sentences, each with 4 references, from the LDC2010E30 corpus, initially released under the DARPA GALE program.

All MT experiments are optimized with MIRA (Crammer et al., 2006) to maximize (T -B)/2.

We test on four different test sets: GALE-web test set from LDC2010E30 corpus (1239 sentences, 4 references), P1R6-web test set from LDC2012E124 corpus (1124 sentences, 1 reference), NIST MT08 newswire portion (691 sentences, 4 references), and NIST MT08 web portion (666 sentences, 4 references).

6.2 Results

Table 2 shows all results of single and joint decoding systems. The B score of the single hiero baseline is 39.2 on MT08-news, showing that it is a strong system. The single F2S baseline achieves comparable scores on all four test sets.

Then, for reference, we present results of joint Hiero and T2S decoding, which is, to our knowledge, a strong and competitive reimplementaion of the work described by Liu et al. (2009). Finally, we present results of joint decoding of hiero and F2S in three interaction schemes: generalization, specification, and interchange.

All three combination schemes significantly improve results of any single system on all four testsets. On average and measured in (T -B)/2, our systems improve the best single system by 0.4 (generalization), 0.7 (specification), and 0.6 (interchange).

The specification comes out as the strongest interaction scheme, beating the second interchange on 2 testsets by 0.1 and 0.4 (T -B)/2 points and on 3 testsets by 0.2 B points.

6.3 Discussion of Results

Interpretations of model behavior with thousands of parameters that may possibly overlap and interfere should be always attempted with caution. In this section we highlight some interesting observations, ac-

Specification		Generalization		Interchange			
X → ∗		∗ → X		X → ∗		∗ → X	
VP	0.069	QP	0.057	*VV*	0.062	*NN*	0.048
IP	0.059	PP	0.054	**VP**	0.044	PP	0.041
VV	0.053	*NN*	0.048	*NN*	0.034	CP	0.035
NR	0.032	DP	0.044	QP	0.025	LCP	0.035
ADVP	0.025	*NR*	0.034	ADVP	0.022	*DEG*	0.031
QP	0.023	DNP	0.032	LCP	0.021	DP	0.028
CC	0.017	NP	0.030	NP	0.018	*DEC*	0.027
DVP	0.017	*LC*	0.025	*P*	0.017	QP	0.027
NP	0.017	*DEC*	0.023	**IP**	0.016	*LC*	0.021
P	0.012	*DEG*	0.023	*NR*	0.016	NP	0.019
...	...	...	...	...	...	...	...
CS	-0.005	*VV*	-0.010	VSB	-0.004	FLR	-0.006
CP	-0.007	PRN	-0.011	*PN*	-0.004	DVP	-0.009
AD	-0.011	*PN*	-0.013	*PU*	-0.004	*BA*	-0.010
VRD	-0.012	*BA*	-0.015	*M*	-0.007	*JJ*	-0.011
PU	-0.028	**VP**	-0.015	VRD	-0.014	*AS*	-0.014
ADJP	-0.028	VRD	-0.028	DNP	-0.023	VRD	-0.017
DNP	-0.045	*JJ*	-0.035	ADJP	-0.039	ADVP	-0.021
PP	-0.064	*VC*	-0.037	PP	-0.058	*PN*	-0.033
PRN	-0.069	DFL	-0.054	DP	-0.070	DFL	-0.038
DP	-0.092	*PU*	-0.073	PRN	-0.080	*PU*	-0.103

Table 3: Examples of specification, generalization, and interchange weights. POS tags in italics.

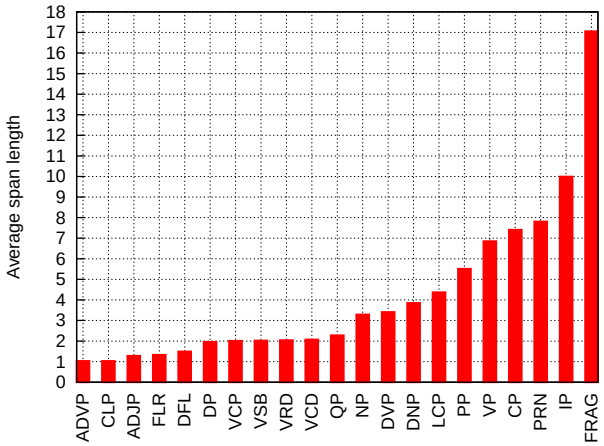


Figure 4: Average span length for selected syntactic labels on GALE-web test set.

#Interactions	Generalization	Inter. gen.
F2S → glue	5557	4202
F2S → hiero	695	1178
total gen.	6252	5380
	Specification	Inter. spec.
phrase → F2S	2763	2235
glue → F2S	946	841
hiero → F2S	683	839
total spec.	4392	3915

Table 5: Rule interactions on GALE-web test set.

companying them with our subjective judgements and speculations.

Table 3 shows the specification and generalization features tuned for the three combination schemes, then sorted by their weights $\lambda_{X\to Y}$ or $\lambda_{Y\to X}$. Features shown at the top of the table are very expensive (the system tries to avoid them), while inexpensive features are at the bottom (the system is encouraged to use them).

The most expensive interactions for the **specification** belong to constituents (IP, VP) that usually occur higher in a syntactic tree (see Figure 4 for average span lengths of selected syntactic labels), and often require non-local reorderings. This indicates that the decoder is discouraged from switching from hiero into F2S derivation at these higher-level spans.

rule type	Generalization		Specification		Interchange	
F2S	18,807	58%	19,399	70%	18,400	61%
Hiero	3,730	12%	2,330	8%	3,133	10%
Glue	7,367	23%	571	2%	4,714	16%
Phrase	2,274	7%	5,484	20%	3,868	13%
total	32,178		27,784		30,115	

Table 4: Rule counts on GALE-web test set.

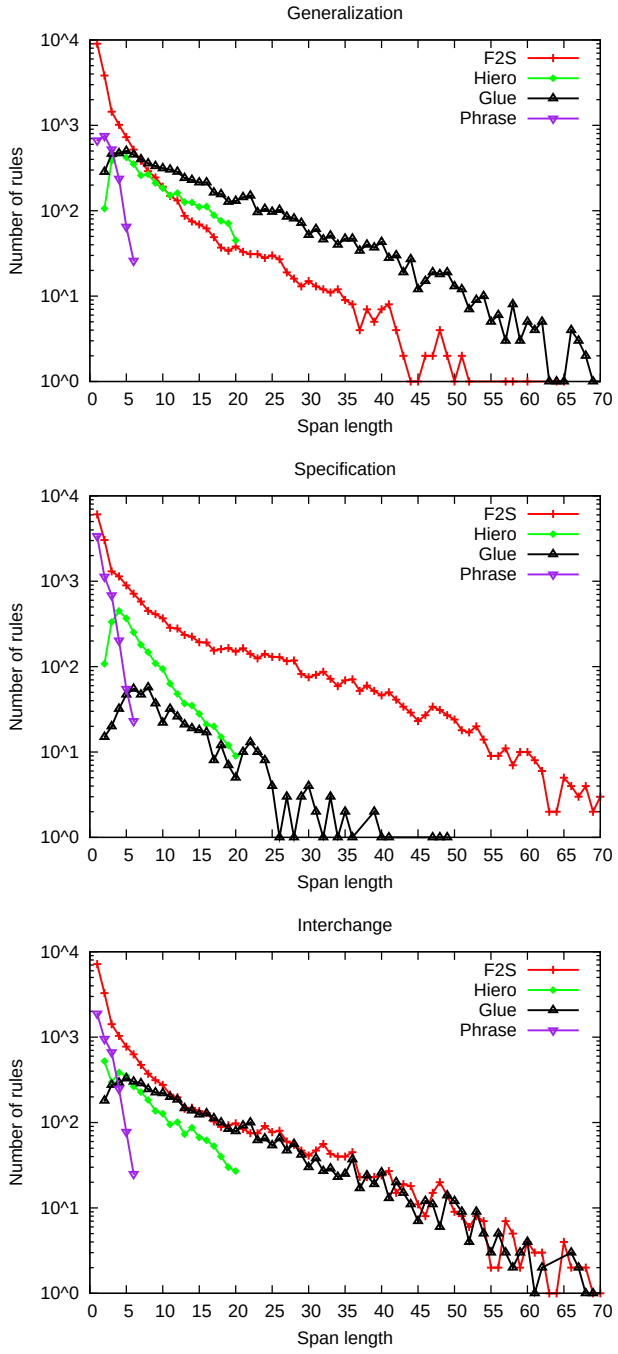


Figure 5: Rule distributions on GALE-web test set.

The third most expensive feature belongs to a part-of-speech tag—the preterminal VV. We may hypothesize that it shows the importance of lexical information for the precision of reordering typically carried out within (parent) VP nodes, and/or the importance of POS information for succesful disambiguation of word senses in translation. Ideally, the system can use a VP rule with a lexicalized VV. Less preferably, the VV part has to be translated by another T2S rule (losing the lexical constraint). In the worst case, the system has to use a hiero hypothesis to translate the VV part (losing the syntactic constraint), risking imprecise translation, since the hiero rule is not constrained to senses corresponding to the source POS VV. Again, the high penalty discourages from using the hiero derivation in this context.

On the other hand, the bottom of the table shows labels that encourage using hiero–DP, PP, DNP, ADJP, etc.–shorter phrases that tend to be monotone and less ambiguous.

Similar interpretations seem plausible when examining the **generalization** experiment. Expensive features related to preterminals (NR, NN, CD) may suggest two alternative principles: First, using F2S rules for thes POS categories and then switching to hiero is discouraged, since these contexts are more reliably handled by hiero due to better lexical coverage and common adjacency in nominal categories. Second, since there is only one attempt to switch from F2S derivation to hiero, letting F2S complete even larger spans (and maybe switching to hiero later) is favorable.

The tail of generalization feature weights is more difficult to interpret. The discount on VP encourages decoder to use F2S for entire verb phrases before switching to hiero, on the other hand, other verb-related preterminals occupy the tail as well, hurrying into early switching from F2S to hiero.

Finally, the feature weights tuned for the **interchange** experiment are divided into two subcolumns. Both generalization and specification weights show similar trends as in the previous two interaction schemes, although blurred (VP and IP descending from the absolute top). Since transitions in both ways are allowed, the search space is bigger and the system may behave differently. It is even possible for a path in the hypergraph to zigzag between F2S and hiero nodes to collect interaction discounts, "diluting" the syntactic homogeneity of the hypothesis.

Figure 5 and Tables 4 and 5 show rule distributions, total rule counts, and numbers of interactions of different types for the three interaction schemes on the GALE-web test set. The scope of phrase rules is limited to 6 words. The scope of hiero rules is limited to 20 words by the commonly used maxParse parameter, leaving longer spans to the glue rule.

The trends of F2S and glue rules show the most obvious difference. In the generalization, F2S rules translate spans of up to 50 words. Glue rules prevail on spans longer then 7 words. The specification is reversed, pushing the longest scope of hiero and glue rules down to 40 words, completing the longest sentences entirely with F2S. The interchange comes out as a mixture of the previous two trends.

All three schemes prefer using F2S rules at shorter spans, to the contrary of our original assumption of phrasal and hiero rules being stronger on local contexts *in general*. Here we may refer again to the specification feature weights for preterminals VV, NR, CC and P in Table 3 and to our previously stated hypothesis about the importance of preserving lexical and syntactic context.

Hiero rules usage on longer spans drops fastest for specification, slowest for generalization, and in between for interchange.

It is also interesting to notice the trends on very short spans (2–4 words) shown by rule distributions and reflected in numbers of interaction types. While specification often transitions from a single phrase rule directly into F2S, the interchange has relatively higher counts of hiero rules, another sign of the hiero and F2S interaction.

Synthesizing from several sources of indications is difficult, however, we arrive at the conclusion that joint decoding of hiero and F2S significantly improves the performance. While the single systems show similar performance, their roles are not balanced in joint decoding. It seems that the role of hiero consists in enabling F2S in most contexts.

We have focused on three special cases of interaction. We see a great potential in further studies of other schemes, allowing more flexible interaction than simple specification, but still more constrained than the interchange. It seems also promising to refine the interaction modeling with features taking into account more information than a single syntactic label, and to explore additional ways of parameter estimation.

7 Conclusion

We have proposed *flexible interaction of hypergraphs* as a novel technique combining hiero and forest-to-string translation models within one decoder. We have explored three basic interaction schemes—*specification*, *generalization*, and *interchange*—and described soft constraints controlling the interactions. We have carried out experiments on large training data and with strong baselines. Of the three schemes, the specification shows the highest gains, achieving improvements from 0.5 to 0.9 (T -B)/2 points over the best single system. We have conducted a detailed analysis of each system output based on different indications of interactions, discussed possible interpretations of results, and finally offered our conclusion and proposed future lines of research.

Acknowledgments

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

References

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of HLT-NAACL*, pages 218–226.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL*, pages 263–270, Ann Arbor, Michigan, June.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.

John Denero, Shankar Kumar, Ciprian Chelba, and Franz Och. 2010. Model combination for machine translation. In *In Proceedings NAACL-HLT*, pages 975–983.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proceedings of HLT-NAACL*, pages 273–280.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of COLING-ACL*, pages 961–968, Sydney, Australia, July.

Xiaodong He, Mei Yang, Jianfeng Gao, Patrick Nguyen, and Robert Moore. 2008. Indirect-HMM-based hypothesis alignment for combining outputs from machine translation systems. In *Proceedings of EMNLP*, pages 98–107, October.

Liang Huang, Kevin Knight, and Aravind Joshi. 2006. Statistical syntax-directed translation with extended domain of locality. In *Proceedings of AMTA*, pages 66–73.

Philipp Koehn, Franz Joseph Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL*, pages 127–133.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of COLING-ACL*, pages 609–616.

Yang Liu, Haitao Mi, Yang Feng, and Qun Liu. 2009. Joint decoding with multiple translation models. In *Proceedings of ACL-IJCNLP*, pages 576–584, August.

Haitao Mi and Liang Huang. 2008. Forest-based translation rule extraction. In *Proceedings of EMNLP*, pages 206–214.

Haitao Mi, Liang Huang, and Qun Liu. 2008. Forest-based translation. In *Proceedings of ACL: HLT*, pages 192–199.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of HLT-NAACL*, pages 404–411.

Antti-Veikko Rosti, Spyros Matsoukas, and Richard Schwartz. 2007. Improved word-level system combination for machine translation. In *Proceedings of ACL*, pages 312–319, Prague, Czech Republic, June.

Taro Watanabe and Eiichiro Sumita. 2011. Machine translation system combination by confusion forest. In *Proceedings of ACL 2011*, pages 1249–1257.

Hui Zhang, Min Zhang, Haizhou Li, and Chew Lim Tan. 2009. Fast translation rule matching for syntax-based statistical machine translation. In *Proceedings of EMNLP*, pages 1037–1045, Singapore, August.