

MTRG presentation (5 12 Mar 2008)

*Chunk-level reordering of source lang. sentences
with automatically learned rules
for statistical machine translation*

Yuqi Zhang
Richard Zens
Hermann Ney

NAACL/HLT 2007, SSST/AMTA workshop

presented by Jeremy G. Kahn

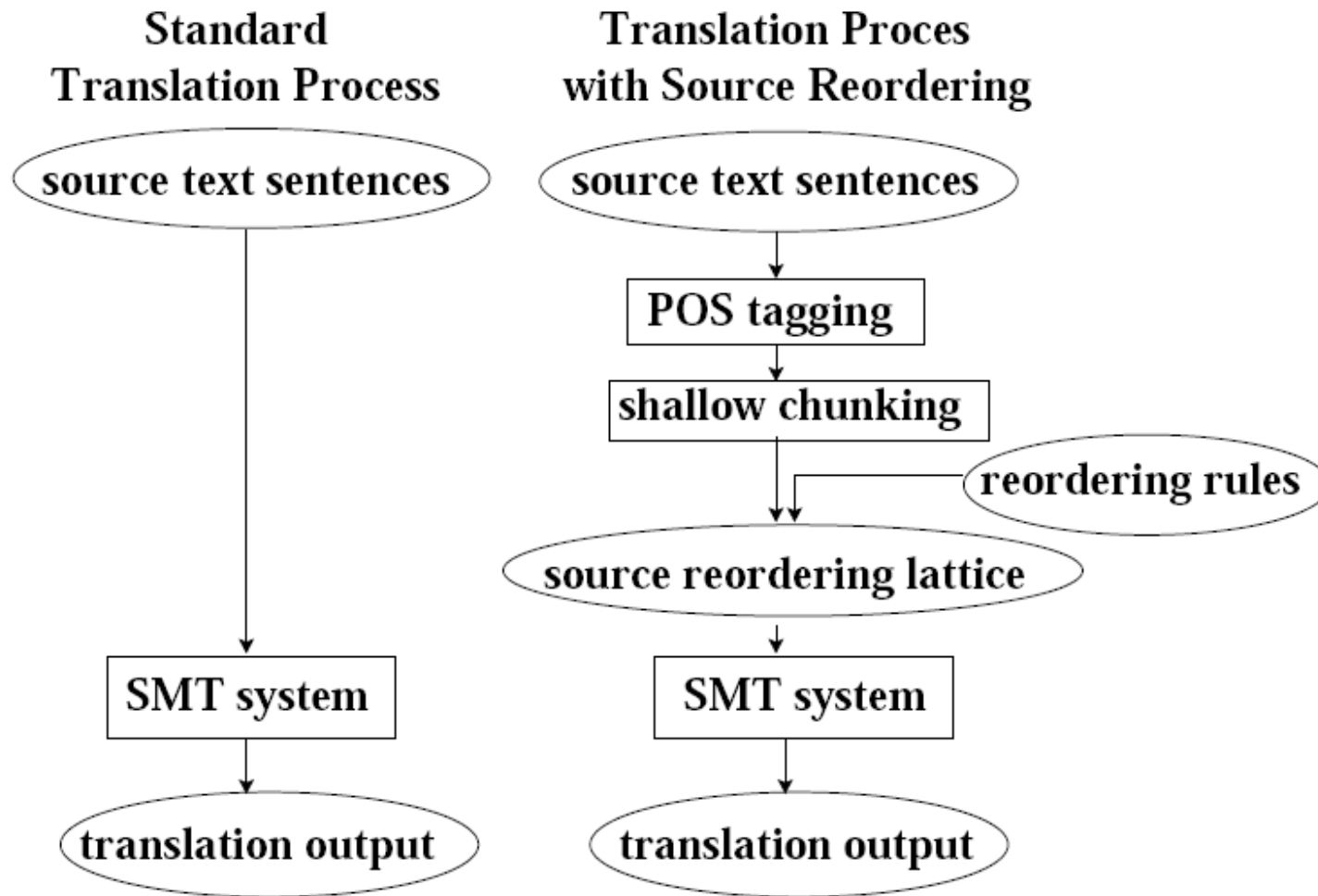
Syntax in MT [for ordering]

- Source pre-ordering
 - learned tree-tree reorderings [Xia & McCord '04]
 - hand-written source-tree rewrites [Collins et al. '05]
 - “Pre-translate” on P OS tags: [Costa-jussà & Fonollosa '06]
- In decoder
 - source coherence [Quirk et al.]
 - target tree structure [Knight et al.] and others
- Reranking
 - Syntactic bi-tree ordering feats indicate good candids [Chen et al. '06, Crego & Mariño '06]

Have cake, will eat too

- Chunk source
 - Shallow syntactic parse (no hierarchy)
 - Tag sequence = {POS tag|chunk tag}+
- Reorder source
 - Learn rules against chunk-tag sequence
 - But don't make a hard decision: reorder into *source* lattice, allowing non-reordered input as well
 - Add additional LM score $p(S')$

System diagram



Chunker details

- Using ICTCLAS POS tagger
- Train YASMET on CTB chunks (first non-unary branch)
- 24 types of chunks, trained on 106K chunk exemplars (487K words)
- 74.5% per-word accuracy, 63.3% per-chunk F

Reordering rule extraction

- GIZA++ intersective alignments
- Merge into source-side chunks
- “Phrase” extraction, discarding cross-phrases
- All other chunk-to-word phrases are rule [templates], with monotonizing.

Reordering rules

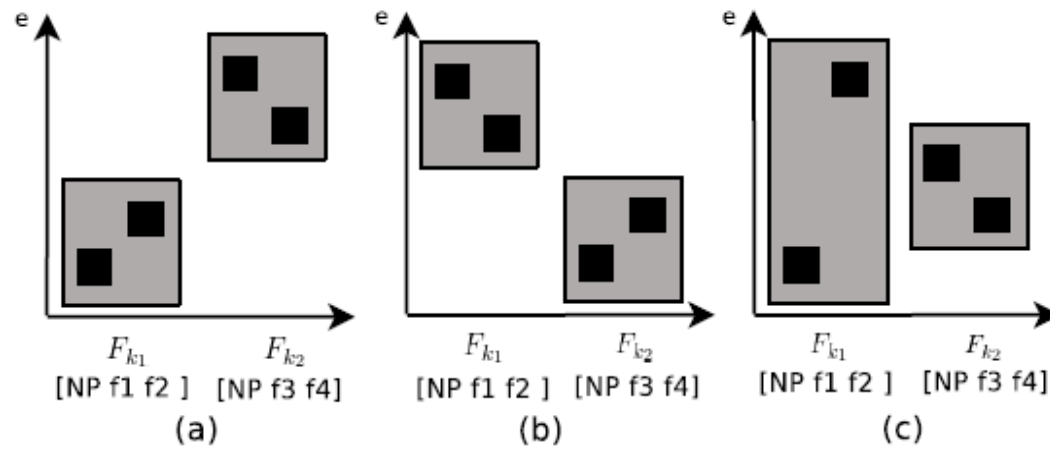
[NP 上海 浦东] [NP开发 与 法制 建设] 并存/v

f0	f1	f2	f3	f4	f5	f6
NP	NP	#	0	1		
NP	NP	#	1	0		
		NP	v	#	0	1
		NP	v	#	1	0
NP	NP	v	#	0	1	2
NP	NP	v	#	1	2	0
NP	NP	v	#	2	0	1

Sentence Permutations

0	1	2	3	4	5	6
2	3	4	5	0	1	6
0	1	2	3	4	5	6
0	1	6	2	3	4	5
0	1	2	3	4	5	6
2	3	4	5	6	0	1
6	0	1	2	3	4	5

Cross phrases



Decoding

- Decoder, LM, TM not changed
 - No retraining of TM!
- Usual log-lin combination:
 - Phrase & word TM, phrase count
 - Phrase-length & word-length feats, distortion model
 - Word TM, 6gm $p(T)$ LM (target)
 - $p(S')$ *reordered-source probability*
- $p(S')$ is trained on monotonized source chunks

Experimental setup

- IWSLT{04,05,06} task: Basic Traveling Expression Corpus (BTEC)
 - {16,16,7} refs per utterance(!)
 - BLEU, NIST, WER, PER reporting
 - Note: chunker out-of-domain
- Trained LM, TM, $p(S')$ model on same corpus:
 - 40k sent train
 - 489 sent dev
 - {500,506,500} sent test

Experimental results

- Baseline: non-monotone
- Source-reordering (Chunk+POS rules)

Table 5: Translation performance for the Chinese-English IWSLT task

		WER[%]	PER[%]	NIST	BLEU[%]
IWSLT04	baseline	47.3	38.2	7.78	39.1
	source reordering	46.3	37.2	7.70	40.9
IWSLT05	baseline	45.0	37.3	7.40	41.8
	source reordering	44.6	36.8	7.51	42.3
IWSLT06	baseline	67.4	50.0	6.65	22.4
	source reordering	65.6	50.4	6.46	23.3
	source reordering+non-monotone decoder	66.5	50.3	6.52	22.4

Improvements on BLEU, WER, but hurts NIST (!)

Using non-monotone decoder *and* source reordering hurts

Experimental results (2)

- Using POS-only vs. POS+Chunk

Table 6: Translation performance of reordering methods on IWSLT 2004 test set

	WER [%]	PER [%]	NIST	BLEU [%]
Baseline	47.3	38.2	7.78	39.1
POS	46.9	37.5	7.38	39.7
Chunk	46.3	37.2	7.70	40.9

- (POS+)Chunk better on all measures
 - How much was re-trained?

Final win: speed [& size]

- On 2006 IWSLT test set, decoding time:
 - Baseline: 17.5 min
 - Source-reordering: 12.3 min(unclear whether cost of reordering is included)
- Size wins on IWSLT-2004:

Table 7: Lattice information for the Chinese-English IWSLT 2004 test data

	avg. density pro sent	used rules	translation time [min/sec]
POS	15.7	6 868	7:08
Chunk	8.2	3 685	3:47

Unclear areas

- $p(S')$ score is learned from “reordered text”-- are all reorderings applied? (I think so!)
- Comparison vs. POS-only source reordering: is $p(S')$ model recomputed?
- Speed comparison: reordering cost?

Questions

- Poor syntactic chunking gives win – how key is it that syntax used at all?
- What are formal differences between source-lattice reordering and decoder-reordering?
 - Source vs. target re-ordering
 - Search constraints

Extensions

- Better chunkers
- Rather than $p(S')$ per path, lattice could include (trainable) weight for particular reordering rule
- What linguistics are actually useful?
 - vs. random chunking
 - vs. high-MI chunking
 - vs. better chunking