CS 294-5: Statistical Natural Language Processing

Machine Translation III

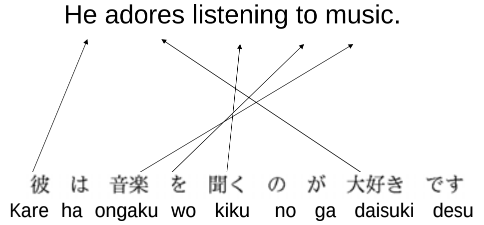
Dan Klein

includes slides from Yamada, Knight, Koehn

Assignment 2 Honors

Why Syntactic Translation?

He adores listening to music.



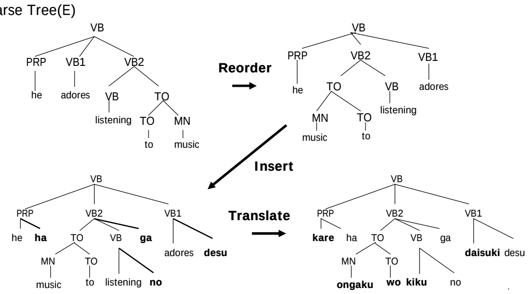
From Yamada and Knight (2001)

Two Places for Syntax?

- Language Model
  - Can use with any translation model
  - Syntactic language models seem to be better for MT than ASR (why?)
  - Not thoroughly investigated [Charniak et al 03]
- Translation Model
  - Can use any language model
  - Linear LM can complement a tree-based TM (why?)
  - Also not thoroughly explored [Yamada and Knight 01]

Parse Tree (E) → Sentence (J)

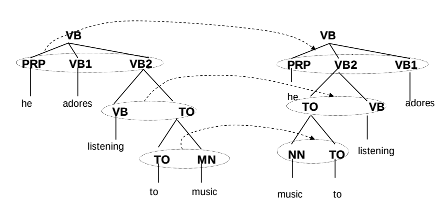
Parse Tree(E)



Sentence(J)

---

1. Reorder



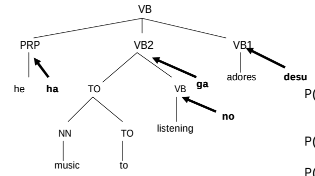
P(PRP VB1 VB2 → PRP VB2 VB1) = 0.723
P(VB TO → TO VB) = 0.749
P(TO NN → NN TO) = 0.893

Conditioning Feature = Child label Sequence

Parameter Table: Reorder

Original Order	Reordering	P(reorder|original)
PRP VB1 VB2	PRP VB1 VB2	0.074
	PRP VB2 VB1	**0.723**
	VB1 PRP VB2	0.061
	VB1 VB2 PRP	0.037
	VB2 PRP VB1	0.083
	VB2 VB1 PRP	0.021
VB TO	VB TO	0.107
	TO VB	**0.893**
TO NN	TO NN	0.251
	NN TO	**0.749**

2. Insert



P(none|TOP-VB) = 0.735

P(right|VB-PRP)* P(ha) = 0.652 * 0.219

P(right|VB-VB) * P (ga) = 0.252 * 0.062

P(none|TO-TO) = 0.900

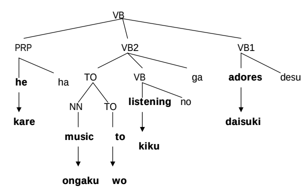
Conditioning Feature = Parent Label & Node Label (position)
none (word selection)

Parameter Table: Insert

Parent label node level	TOP VB	VB VB	VB TO	TO TO	TO NN	TO NN
P (none)	**0.735**	0.687	0.344	**0.700**	**0.900**	**0.800**
P (left)	0.004	0.061	0.004	0.030	0.003	0.096
P (right)	0.260	**0.252**	**0.652**	0.261	0.097	0.104

W	P (insert-w)
ha	**0.219**
ta	0.131
wo	0.099
no	**0.094**
ni	0.080
te	0.078
ga	**0.062**
⋮	⋮
desu	**0.0007**
⋮	⋮

3. Translate



P (he → kare) = 0.952
P (music → ongaku) =0.900
P (to → wo) = 0.038
P (listening → kiku) = 0.333
P (adore → daisuki) = 1.000

Conditioning Feature= word (E) identity

Parameter Table: Translate

E	adores	he	listening	music	to
J	**daisuki 1.000**	**kare 0.952** NULL 0.016 nani 0.005 da 0.003 shi 0.003 ⋮ ⋮	**kiku 0.333** kii 0.333 mi 0.333	**ongaku 0.900** naru 0.100	ni 0.216 NULL 0.204 **to 0.133** no 0.046 wo 0.038 ⋮ ⋮

Note: Translation to NULL = deletion

Experiment: Y+K 03

- Training Corpus: J E2K sentence pairs
- J: Tokenized by Chasen [Matsumoto, et al., 1999]
- E: Parsed by Collins Parser [Collins, 1999]
  --- Trained: 40K Treebank, Accuracy: ~90%
- E: Flatten parse tree
  --- To Capture word-order difference (SVO->SOV)
- EM Training: 20 Iterations
  --- 50 min/iter (Sparc 200Mhz 1-CPU) or
  --- 30 sec/iter (Pentium3 700Mhz 30-CPU)

Result: Alignments

	Ave. Score	# perf sent
Y/K Model	0.582	10
IBM Model 5	0.431	0

- Ave. by 3 humans for 50 sents
- okay(1.0), not sure(0.5), wrong(0.0)
- precision only

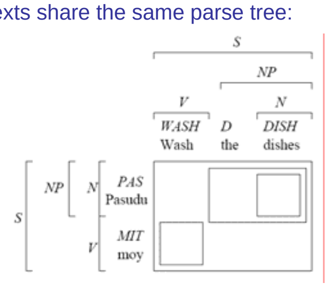
Result: Alignment Example

Syntax-based Model

IBM Model 3

Synchronous Grammars

- Multi-dimensional PCFGs (Wu 95, Melamed 04)
- Both texts share the same parse tree:



Synchronous Grammars

- Formally: have paired expansions

$S \rightarrow NP\ VP$ $VP \rightarrow V\ NP$

$S \rightarrow NP\ VP$ $VP \rightarrow NP\ V$

- … with probabilities, of course!
- Distribution over tree pairs
- Strong assumption: constituents in one language are constituents in the other
- Is this a good assumption? Why?

m	1	2	3	4	5	6	7	8	9	10	11	12
%	100	100	100	99.04	94.83	86.07	73.35	58.51	43.70	30.62	20.18	12.55

Details

- Distinctions in lines of work are in the details:
  - What about insertions?
  - What about deletions?
  - How flat can rules be?
  - Multiple transductions of rules?
- Recent work (Eisner 04, Melamed 04) much more flexible than early work
  - … but still no killer results