Linguist vs. Machine: Rapid Development of Finite-State Morphological Grammars

University of Colorado **Boulder**

Australian National University

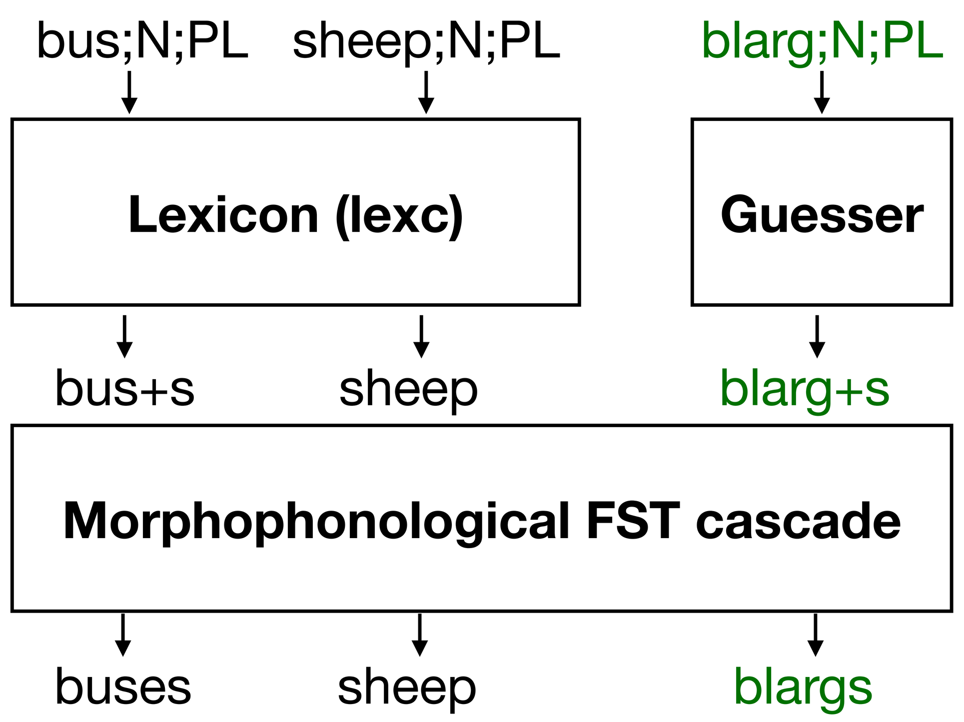
Sarah Beemer, Zak Boston, April Bukoski, Daniel Chen, Princess Dickens, Andrew Gerlach, Torin Hopkins, Parth Anand Jawale, Chris Koski, Akanksha Malhotra, Piyush Mishra, Saliha Muradoğlu, Lan Sang, Tyler Short, Sagarika Shreevastava, Elizabeth Spaulding, Tetsumichi Umada, Beilei Xiang, Changbing Yang, Mans Hulden

`first.last@colorado.edu`

`https://github.com/mhulden/7565tools`

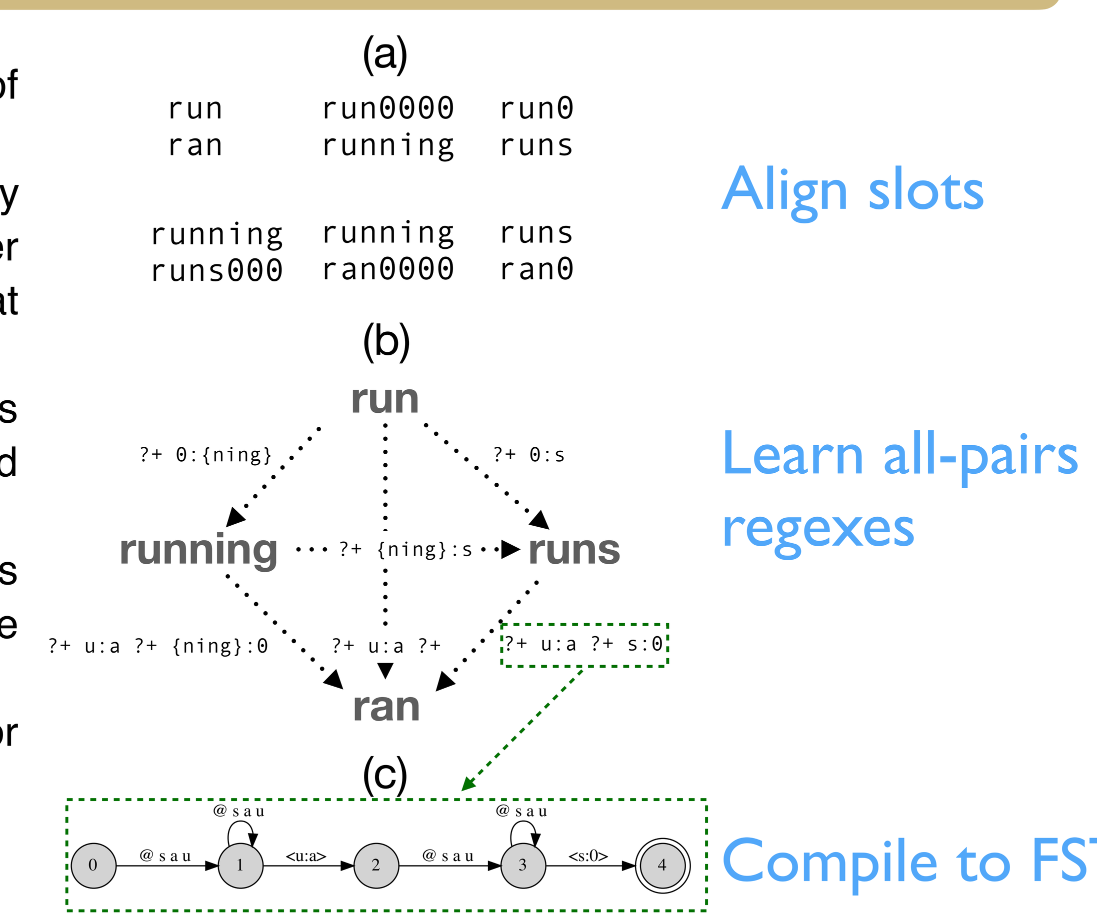
(1) Hand-written FST grammars

- Evaluate effort required to develop FST-grammars that exceed seq2seq models in accuracy
- A team of 20 with linguistic training and training in FST tools did rapid development of 25 languages with the foma finite-state tool
- Linguists develop grammars based on training/dev sets
- Performance equal to best neural model in task on **11** languages and significantly better on **2** (Ingrian, Tagalog)
- TL;DR: only saw improvement vs. seq2seq models with languages with complex inflectional classes and complex morphophonology



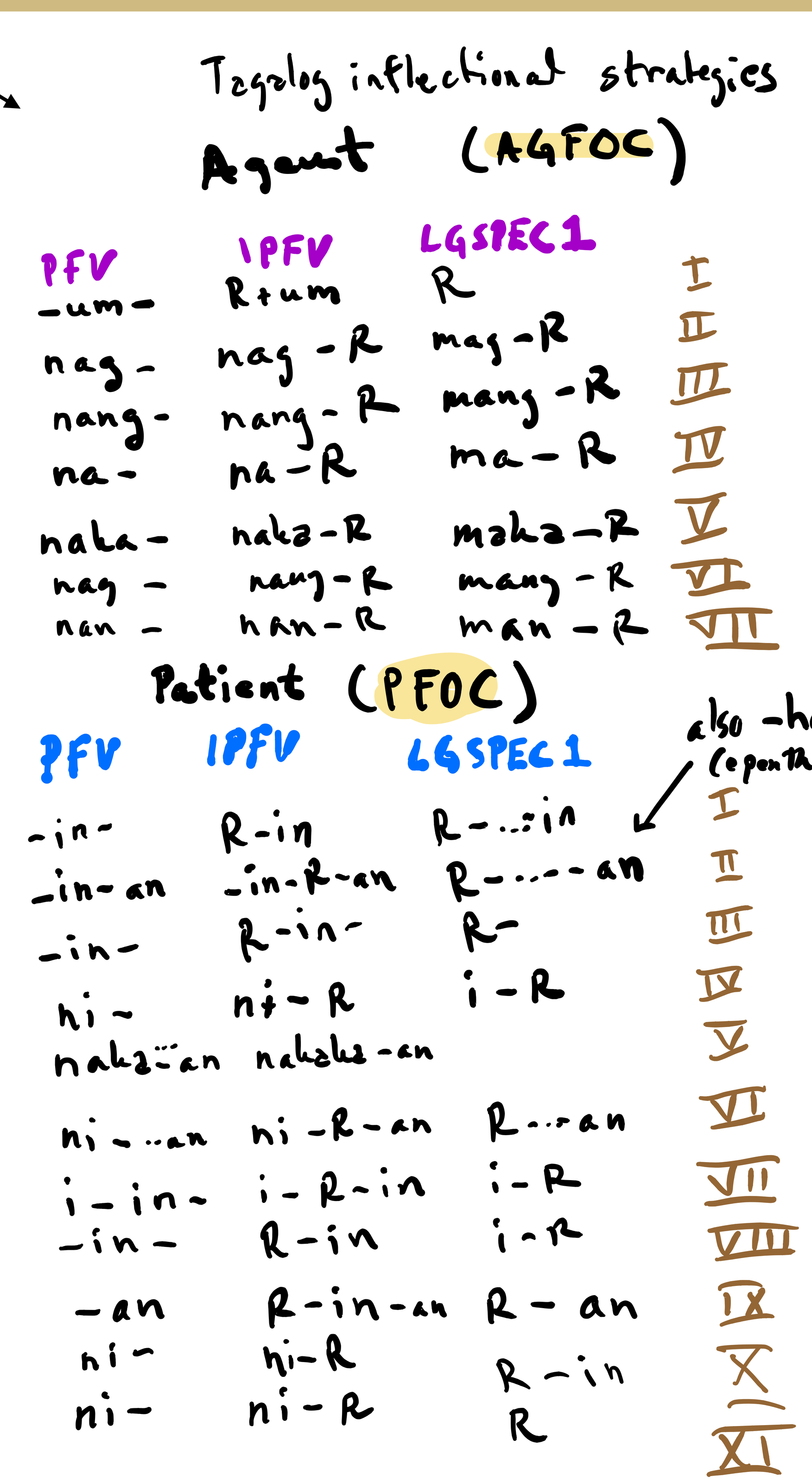
(2) Non-neural inflection model and inflectional class clustering

- Also developed various tools to aid rapid development and analysis of inflectional behavior
- A non-neural model for filling partially filled missing paradigms by creating simple FSTs that inflect each known slot from every other known slot by learning regular expressions that encode an FST that does this
- This can be used to solve the task by generating candidates for slots from all known slot-slot FST transformations for other lexemes and using them in a voting scheme for the lexeme at hand (see fig below)
- It can also be used for clustering lexemes into inflectional classes (helpful for developing initial hypotheses about classes when large numbers of partial paradigms are available)
  - The number of identical slot-to-slot transformation FSTs for each lexeme is used as a distance measure for clustering



example

Paper-and-pencil linguistics



Results

tst[1] = handwritten (1)
tst[2] = learned (2)

Language	trn[1]	dev[1]	tst[1]	tst[2]
aka	100.0	100.0	**100.0**	89.8
ceb	85.2	86.2	86.5	84.7
crh	97.5	97.0	96.4	97.7
czn	79.0	76.0	72.5	76.1
dje	100.0	100.0	**100.0**	**100.0**
gaa	100.0	100.0	**100.0**	**100.0**
izh	93.4	91.1	**92.9**	77.2
kon	100.0	100.0	98.7	97.4
lin	100.0	100.0	**100.0**	**100.0**
mao	85.5	85.7	66.7	57.1
mlg	100.0	100.0	**100.0**	-
nya	100.0	100.0	**100.0**	**100.0**
ood	81.0	87.5	71.0	62.4
orm	99.6	100.0	**99.0**	93.6
ote	91.2	93.5	90.9	91.3
san	88.5	89.7	89.0	88.3
sna	100.0	100.0	**100.0**	99.3
sot	100.0	100.0	**100.0**	99.0
swa	100.0	100.0	**100.0**	**100.0**
syc	89.3	87.3	88.3	89.1
tgk	100.0	100.0	**93.8**	**93.8**
tgl	77.9	75.0	**77.8**	-
xty	81.1	80.0	81.7	70.3
zpv	84.3	77.9	78.9	81.1
zul	82.9	88.1	83.3	88.5

Filling in missing forms and clustering example

MacGyver	abominate	render	
MacGyvering	1 abominating	3 rendering	V.PTCP;PRS
?	abominated	rendered	V.PTCP;PST
-	2 -	4 -	V;NFIN
MacGyvers	abominates	renders	V;SG;3;PRS

```
Candidates for ?: [MacGyvered, MacGyverd, MacGyvered, MacGyvered]
                   1           2          3           4
```