

OT and JXNL-Soar

Nathan Glenn

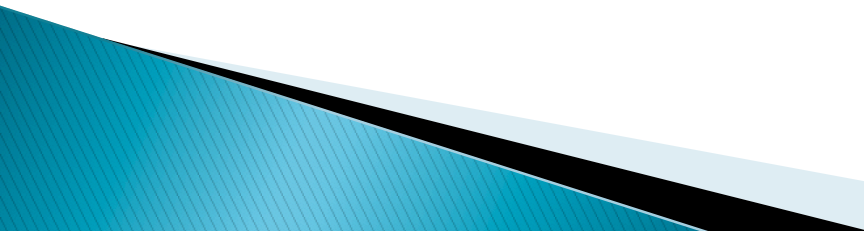
Nathan.g.glenn@gmail.com

BYU

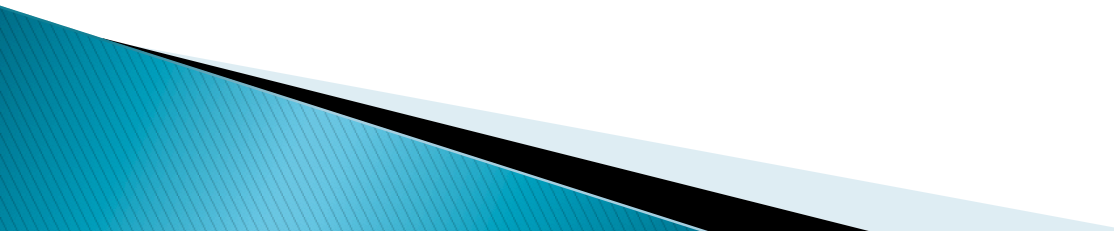
Optimality Theory

- ▶ Is a method of predicting linguistic or other behavior using ranked violable constraints
- ▶ Good for predicting linguistic phenomena, cross-linguistic and dialectic variation
- ▶ Currently biggest use is in phonology
 - Also used for morphology, syntax, etc.
- ▶ Can be combined with other frameworks (Minimalist syntax, Analogical Modeling)

Desired Application

- ▶ Case assignment in Japanese
 - ▶ Possible assignments:
 - は・が–topic/subject
 - を–object
 - に–dative
 - ▶ Usually straightforward, but there are exceptions!
 - ▶ Quirky case verbs allow dative marking on the subject
- 

How OT Works

- ▶ Tableau shows violable constraints and possible behavior
 - ▶ Compare and rule out cells on a column-by-column basis
 - ▶ Index points to outcome
- 

Approach from Woolford's “Case Patterns”

Generalized approach accounting for case patterns in Japanese, Icelandic, and other quirky case languages

input: 思える (intransitive, quirky case)	Faith-lex _{trans}	*dat	Faith- lex	*acc	*nom
subj-dat obj-nom		*!			*
subj-dat obj-acc		*!		*	
☞ subj-nom obj- acc			*	*	

input: 分かる (transitive, quirky case)	Faith-lex _{trans}	*dat	Faith-lex	*acc	*nom
☞ subj-dat obj- nom		*			*
subj-dat obj-acc		*		*!	
subj-nom obj-acc	*!		*	*	

Soar Implementation

- ▶ Initialize `s1 ^tableau.constraint`
 - Each constraint has a `^badness` integer value
- ▶ Match violations and mark operator with `^violation` pointing to `tableau.constraint`
- ▶ 1 production does all the logic
 - Find highest violation that `<o1>` violates which `<o2>` doesn't
 - If `<o2>` doesn't violate anything with higher badness, prefer `<o2>`

Nuggets

- ▶ General OT implementation
- ▶ Natural use in Soar
- ▶ Good account for Japanese case assignment
- ▶ Use of OT is non-exclusive

Coal

- ▶ Not integrated into JXNL-Soar yet
- ▶ Japanese WordNet has no verb frames (argh!)