Using Optimality Theory to Guide Surface Realization

Corinna Anderson

Project supervised by
Claire Gardent

Introduction and goals

Apply Optimality Theory to existing NLG system

Current GenI output = unordered list of grammatical paraphrases for each semantic input (from 2 to over 500 paraphrases per input)

What we want to do

- Main goal: Identify default output in generation
  - allow determinism in generation, but preserve flexible generative capacity for paraphrases
- Secondary goal: Order other output paraphrases according to markedness
- System should be linguistically principled and apply to all inputs

How we do it

- Apply constraints on syntactically marked structures
- Identify & exploit linguistic generalizations in grammar design

Background & resources

GenI surface realizer & grammar resources (TAG, XMG)

- Associates NL expression with syntax and semantics
- Reversible for parsing and generation
    - Parsing: syntax → semantics
    - Generation: semantics → syntax
- Generates multiple grammatical paraphrases for each input

Optimality Theory: constraint-based framework

identifies the optimal output for every input

constraints ranked within a given grammatical system

Tree Adjoining Grammar…

- Syntactic trees for full sentences are produced by combinations of elementary TAG trees
- Lexicalized TAG: elementary trees anchored to words

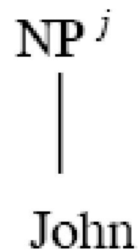
…with XMG: eXtensible MetaGrammar

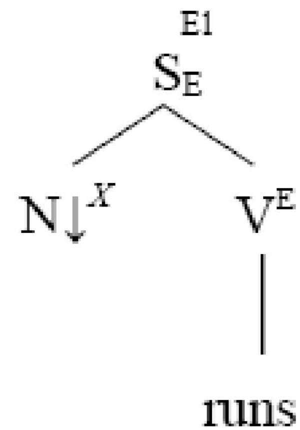
- Grammar derived from XMG specification = set of statements describing elem. trees + associated semantics
- Result: Each tree in the grammar is associated with a set of classes representing syntactic properties

TAG & XMG in GenI

GenI semantic input `[run(e), agent(e, j), john(j)]`

⇉ Lexicalized TAG elementary trees selected





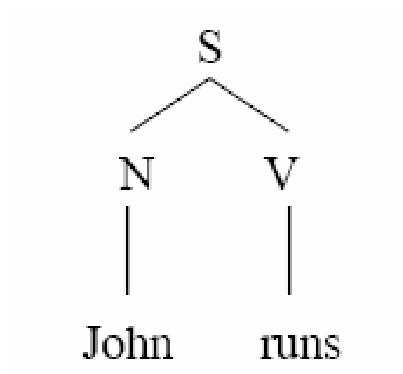
\+

Info associated with TAG elementary trees:

- Semantics `john (j)` `run (e, x)`
- Set of XMG classes `{ProperNoun}` `{CanonicalSubject, ActiveVerbForm}`

TAG & XMG in GenI

Resulting **TAG derived tree** is associated with its **set of sets of XMG classes**



`{{ProperNoun}, { CanonicalSubject, ActiveVerbForm}}`

Important : the XMG classes are also linked with GenI output sentence

Optimality Theory

General design of OT

3 components

- GEN : generation of all possible outputs
- CON : set of universal constraints on output
  with language-specific ranking
- EVAL : evaluation of output candidates
  against constraints with ranking

Result: Identification of optimal output candidate

OT: Theoretical background

- All constraints are universal
- All constraints are violable, many are in conflict
- Cross-linguistic differences due to ranking differences
- Unmarked patterns emerge from constraint interaction

- 2 basic kinds of constraints in OT
  - Faithfulness (input-output relation prohibits deletion /addition of structure)
  - Markedness (prohibitions against marked structures)
- GenI syntax-semantics link ensures input-output faithfulness
- We assume GenI produces only grammatical outputs

→ OT-GenI needs markedness constraints, but not faithfulness

OT tableaux & conventions

- Constraint violations assigned [*] per instance of prohibited structure
- Output candidates violating highly ranked constraints are ruled out
- → Candidate with the fewest & lowest ranked violations “wins”

Toy example : Markedness constraints *X and *Y

2 candidates

Constraint ranking: *X >> *Y

	*X	*Y
Candidate A: XXYY	**	**
→ Candidate B: XYYY	*	***

Integrating OT into GenI

3 components of Optimality Theory:

GEN (generation of candidates): done automatically by GenI

CON (constraints & ranking): manually checked descriptive structural features of paraphrases against individual XMG class names associated with output trees

EVAL (evaluation of candidates):
ot-geni script designed with the help of Eric Kow
(constraint ranking = ordered list of lists in Haskell)
- checks the constraints against the classes
- records the violations into a table for each input

OT-GenI constraint design, part 1

- All constraints are specified in terms of individual XMG classes
- OT framework can provide a full ordering when sufficient syntactic information is accessible to the constraints
- Constraint type 1: simple prohibitions
  (constraints of form *CLASSNAME)
- Let’s see how these look in an example…

OT-GenI constraints in action

Simplified constraint example

La femme à qui Jean ment part 'The woman to whom John lies is leaving'

```
Input semantics: [A:agent(B C) E:agent(F G) D:femme(C) E:le(C) H:jean(G)
   E:mentir(F) A:partir(B) E:patient(F C)]
```

Constraint ranking: *IMPERSONALSUBJECT >> *SUBJECTINVERSION >> *CLEFT

	*IMPERSSUBJ	*SUBJINVERSION	*CLEFT
a. Il part la femme à qui ment jean	*	*	
b. Part la femme à qui jean ment		*	
c. Part la femme à qui ment jean		**	
d. C'est la femme à qui jean ment qui part			*
e. C'est la femme à qui ment jean qui part		*	*
→ f. La femme à qui jean ment part			
g. La femme à qui ment jean part		*	

Sample OT-GenI constraints & what they refer to

Constraint in tableau	refers to XMG Class	Description
*SUBJECT INVERSION	`InvertedNominalSubject`	Subject appears after verb (including sentence-final position)
*PASSIVE	`passiveVerbMorphology`	Passive verb form and syntax
*CLEFT-NON-SUBJECT	`UnboundedCleft`	Non-subject argument X expressed with cleft *c'est X que / dont / etc...*
*CLEFT-SUBJECT	`CleftSubject`	Subject argument X expressed with cleft *c'est X qui*
*DE-AGENT	`dian0V1Passive`	Agent of passive verb expressed with de (rather than par)

Constraint design, part 2

Constraint type 2: Conjunction of XMG classes

Example : Impersonal subjects

Input semantics:

`[arriver(e), agent(e, x), garçon(x), (x)]`

Paraphrases generated:

Un garçon arrive (default ‘A boy arrives/ is coming’)

Il arrive un garçon (‘There arrives/ is coming a boy’)

Problem for constraint design:

A simple constraint against impersonal subjects would punish good outputs for impersonal verbs like *falloir* and *sembler*

cf. paraphrases *Il semble que Marie part (‘It seems that Mary is leaving’)* vs. *Marie semble partir (‘Mary seems to be leaving’)* Both are fine!

Constraint design, part 2

Solution : conjoined constraint type

Combine XMG class for “impersonal subject” and for intransitive verb with a single NP argument (N0v)

`*[ImpersonalSubject, N0v]`

- Conjunction of XMG classes searches for co-occurrence within single elementary tree (only the verb tree can have these classes)

  `{{ ImpersonalSubject, ...., N0v, class1},{class2, class3}}`

Exploits the set-of-sets structure of XMG class information associated with derived trees (full sentences)

→ Violation for *Il arrive un garçon* but not *Il semble qu’un garçon arrive*

Evaluation, part 1

Test suite: inputs of different verb classes (subcat frames) in simple clauses, 1 and 2 levels of clausal embedding, or arguments modified by a relative clause

Evaluation questions:

- Do the constraints and their ranking identify a default?
- Does the same ranking produce a full or partial ordering of all paraphrases, and do native speakers agree?

Positive results

- Unique default output identified for most inputs
- A partial ranking for all outputs was produces with 1-2 paraphrases per rank (examples in slide 18)

A nice side effect

Some instances of overgeneration are ruled out as defaults (if they involve marked structures)

Example : overgeneration of impersonal subjects (lexical)

Input semantics:

`[partir(e), agent(e, m), marie(m)]`

Paraphrases generated:

→ *Marie part* (default ‘Mary leaves’)

vs. *Il part Marie* (questionable ‘There leaves Mary’)

Constraint against impersonal subjects is ranked high → results in ordering far below “optimal” for such outputs

Evaluation, part 2

Some limitations based on grammar features:

Not all differences are associated with a difference in XMG classes – some are due to distinct TAG trees with identical descriptions

XMG classes cannot distinguish between

- Argument order variants
- standard double object constructions (e.g. *donner*)
- all subclasses of control verbs (e.g., *suggérer à paul de partir*)
- verbs with sentential objects (e.g., *dire à qqn que...*)
- *discuter Marie avec Paul* vs. *discuter avec Paul Marie*
- Passives of any of these verb types: *Marie est discutée par Jean avec Paul* and *Marie est discutée avec Paul par Jean*

- Embedded vs. main clause: *Jean demande si c'est Paul qui vient* and *C'est Jean qui demande si Paul vient*.

Conclusion

- Default can be identified using a small number of markedness-type OT constraints
- OT-GenI can provide a full ordering when sufficient syntactic information is accessible to the constraints

Further work:

- Improvement on current goal of default ID & full ordering
  - Modification of what OT constraints can access in the grammar
- Expanding constraint system & input-output link
  - Introduction of features for context (not just semantics)
  - Allow syntactic default to interact with additional constraints on discourse features / information structure : Topic, Focus, etc (i.e. motivation for non-canonical structures interacts with default preferences)

Thanks

special thanks to

Eric Kow

and

Yannick Parmentier

☞ ☞ ☞ ☞ ☞ ☞ ☞ ☞ ☞ ☞

Questions?

OT tableau for "Jean aime Marie"

```
Sentence: jean aime marie (paraphrases-1/verbs t20)

Input Semantics:[A:agent(B C) A:aimer(B) E:jean(C) F:marie(D) A:patient(B D)]
```

Constraint ranking: *DE-AGENT >> *SUBJECT INVERSION >> *PASSIVE >> *CLEFT-DONT >> *CLEFT-NON-SUBJECT >> *CLEFT-SUBJECT

	*DE-AGENT	*SUBJINVERS	*PASS	*CLEFTDONT	*CLEFTNONSUBJ	*CLEFTSUBJ
→ a. Jean aime Marie						
b. C'est Jean qui aime Marie						*
c. C'est Marie que Jean aime					*	
d. Marie est aimée par Jean			*			
e. C'est Marie qui est aimée par Jean			*			*
f. C'est par Jean que marie est aimée			*		*	
g. C'est Marie qu'aime Jean		*			*	
h. C'est par Jean qu'est aimée Marie		*	*		*	
i. Marie est aimée de Jean	*		*			
j. C'est de Jean que Marie est aimée	*		*		*	
k. C'est Jean dont Marie est aimée	*		*	*	*	
l. C'est Marie qui est aimée de Jean	*		*			*
m. C'est de Jean qu'est aimée Marie	*	*	*		*	
n. C'est Jean dont est aimée Marie	*	*	*	*	*	

OT tableau for "Jean aime Marie"

Input Semantics:[A:agent(B C) A:aimer(B) E:jean(C) F:marie(D) A:patient(B D)]

Constraint ranking: *DE-AGENT >> *SUBJECT INVERSION >> *PASSIVE >> *CLEFT-DONT >> *CLEFT-NON-SUBJECT >> *CLEFT-SUBJECT

	*DE-AGENT	*SUBJINVERS	*PASS	*CLEFTDONT	*CLEFTNONSUBJ	*CLEFTSUBJ
→ a. Jean aime Marie						
b. C'est Jean qui aime Marie						*
c. C'est Marie que Jean aime					*	
d. Marie est aimée par Jean			*			
e. C'est Marie qui est aimée par Jean			*			*
f. C'est par Jean que marie est aimée			*		*	
g. C'est Marie qu'aime Jean		*			*	
h. C'est par Jean qu'est aimée Marie		*	*		*	
i. Marie est aimée de Jean	*		*			
j. C'est de Jean que Marie est aimée	*		*		*	
k. C'est Jean dont Marie est aimée	*		*	*	*	
l. C'est Marie qui est aimée de Jean	*		*			*
m. C'est de Jean qu'est aimée Marie	*	*	*		*	
n. C'est Jean dont est aimée Marie	*	*	*	*	*	