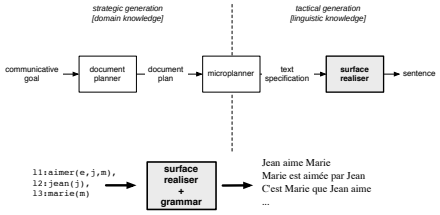
1 Surface realisation

A typical natural language generation pipeline:



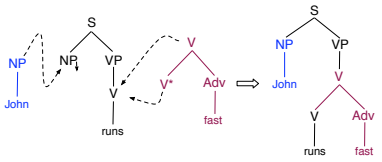
1.1 L_U Flat semantics

1. An L_U formula is a set of literals
   `l1:aimer(e,j,m), l2:jean(j), l3:marie(m)`
2. Each literal consists of a predicate, a label and some arguments:

3. The label and arguments are either constants or unification variables.

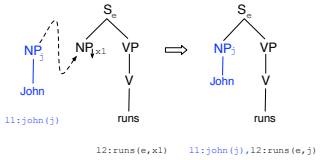
1.2 Feature-Based Lexicalised Tree Adjoining Grammar (FB-LTAG)

An FB-LTAG associates each word with a set of trees.
Two combining operations: substitution and adjunction.



1.2.1 FB-LTAG with an L_U semantics

Each tree is associated with a semantic representation

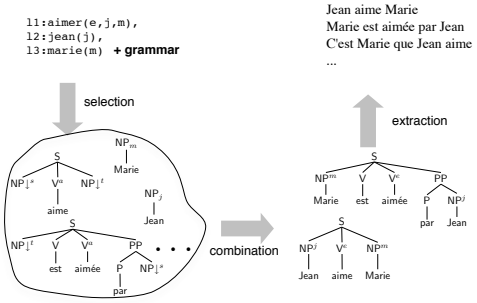


$Sem(t_1 + t_2) = Sem(t_1) \cup Sem(t_2)$ modulo unification

1.2.2 SemFraG

Reversible grammar for French. Will be made available.
For now, contact Claire Gardent `gardent@loria.fr`

1.3 Realisation algorithm



GenI realiser:
`http://trac.loria.fr/~geni`

2 Polarity filtering

2.1 Lexical ambiguity

Lexical ambiguity is the possibility for a literal to be expressed in several ways:

`l0:picture(p)`	`l1:cost(c,p,h)`	`l2:high(h)`
« picture »	« cost of »	« is high »
« painting »	« costs »	« a lot »

The number of lexical combinations is exponential:

$$\prod_{1 \leq i \leq n} a_i$$

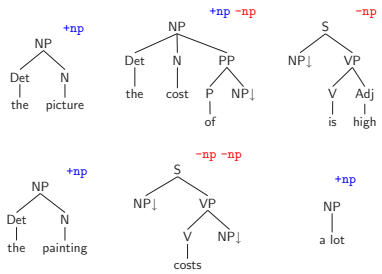
n, number of literals in an input semantics
a_i, ambiguity of the i-th literal

2.2 Syntactic incompatibilities

« picture » « cost of » « is high »	the cost of the picture is high
« picture » « costs » « a lot »	the picture costs a lot
~~« picture » « cost of » « a lot »~~	~~the cost of the picture a lot~~

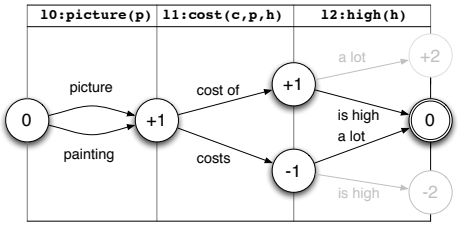
2.3 Polarities

Each lexical entry in the grammar is associated with a set of polarities which represent its syntactic resources and needs.



« picture » « cost of » « is high »	+1np	0np	-1np	(= 0np)
« picture » « costs » « a lot »	+1np	-2np	+1np	(= 0np)
~~« picture » « cost of » « a lot »~~	+1np	0np	+1np	(= +2np)

2.4 Polarity automaton



2.5 An example

L'homme qui discute philosophie avec Paul dit que Jean part.

	no filtering	**filtering**
lexical combinations	2 436 672	4 136
substitutions	26 149	3 284
adjunctions	5 014	630
realisation time (s)	1 615	25

References

[Perrier 2003] Perrier, G. (HDR, 2003) “Les grammaires d’interaction”

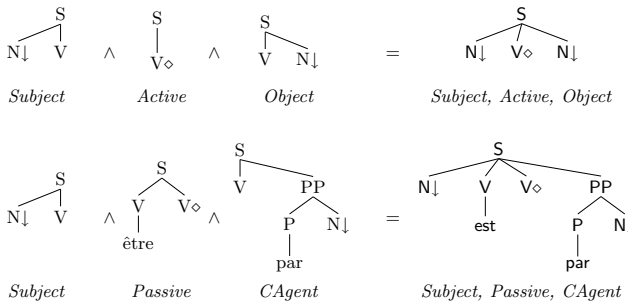
[Kow 2005] Kow, E. (ESSLLI student session, 2005, Edinburgh) “Adapting polarised disambiguation to surface realisation”

3 Paraphrase selection

~~A speech is going to be made by him.~~
He is going to make a speech.

3.1 Metagrammar

Tree fragments (dominance and linear precedence constraints) combine to make TAG trees:



eXtensible MetaGrammar compiler (XMG)
`http://sourcesup.cru.fr/xmg/`

3.2 Tree properties for selection

Enriched input semantics (each literal can be associated with a set of tree properties, to act as filters)

`l0:give(e,x,y,z)[Passive,ToObject],l1:joe(x),l2:sue(y),l3:car(z)`
Joe gives the car to Sue.
~~Sue is given the car by Joe.~~
~~The car is given to Sue by Joe.~~

3.3 Evaluation

`l0:give(e,x,y,z),l1:joe(x),l2:sue(y),l3:car(z)`

⇓

Joe gives the car to Sue.	`Active, Object`
Sue is given the car by Joe.	`Passive, Object`
The car is given to Sue by Joe.	`Passive, ToObject`

For each pair of paraphrases: do they have distinct *enriched* semantics?
Yes, for 98% of pairs (87 cases, or 1528 paraphrases)

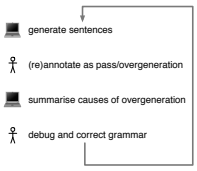
References

[Crabbé and Duchier 2004] Crabbé, B. and Duchier, D. (2004). “Metagrammar Redux”.

[Gardent and Kow 2007] Gardent, C. and Kow, E. (ACL 2007, Prague). “A symbolic approach to near-deterministic surface realisation using Tree Adjoining Grammar”

4 Reducing overgeneration

Incremental, semi-automatic approach:



4.1 Derivations log

For each string: its derivation tree, lexical selection and the tree properties of each lexical entry used.

```
Output: Jean se demande si c’est Paul qui vient
demander:n8 <-(s)- venir
demander:n1 <-(s)- jean
venir:n4 <-(s)- paul

demander TnOClVs1int-630
  CanonicalSubject NonInvertedNominalSubject
  SententialInterrogative
venir TnOV-615
  CleftSubject NonInvertedNominalSubject
paul TproperName-45
jean TproperName-45
```

4.2 Suspects report

For each lemma: TAG families, trees, tree properties which *only* appear in cases of overgeneration.

```
input t90
Lemma: dire
TnOVn1 (all) - InfinitiveSubject Passive
  [699] CanonicalCAgent Passive
  [746] CanonicalGenitive dePassive
  [702] CleftCAgentOne Passive
  [752] CleftDont dePassive
```

Also: combinations of lexical items which only appear in overgeneration.

```
Input t70
consistently overgenerating derivation items
le:Tdet-17:n0 <-(a)- riche:TnOvA-90
```

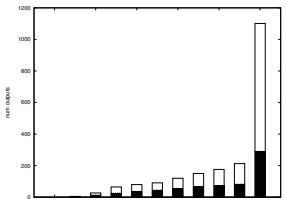
4.3 A quarter of the output

We have eliminated 70% of the strings produced with 13 modifications to the metagrammar (31 lines, 12 hours).

	total	maximum	average	median
before	28000	4900	200	25
after	8400	710	60	12

number of strings per case

We got greater reductions for longer phrases.



References

[Gardent and Kow 2007] Gardent, C. and Kow, E. (ENLG 2007). “Spotting overgeneration suspects”