Unsupervised Methods for Head Assignments

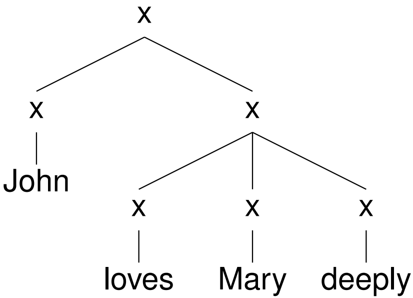
Federico Sangati & Willem Zuidema

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION
University of Amsterdam

April 3, 2009

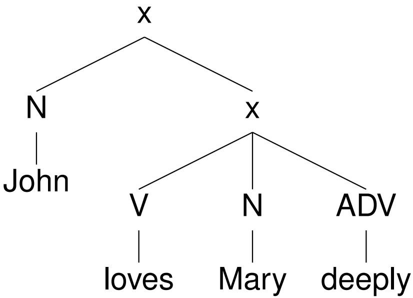
The big picture

Bracketing (e.g., Klein & Manning'04; Seginer'07)



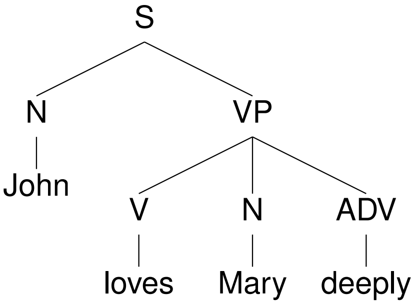
The big picture

POS tagging (e.g., Schütze'93, Chater'95)



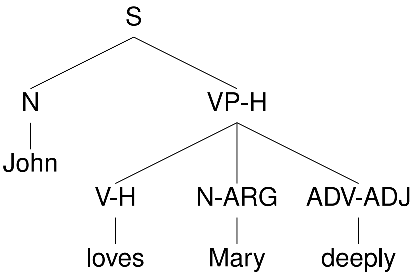
The big picture

Phrase categories (e.g., Borensztajn & Zuidema'07; Reichart & Rappoport'08)



The big picture

Heads and Argument structure



Outiline

1. Introduction
   - Heads in Constituency Structures
2. Assigning heads
   - Rule-based methods
   - LTSG
   - Unsupervised Learning of Heads
3. Evaluations
   - Parsing results
   - Gold standard evaluations
   - Dependency Parsing
4. Final Remarks

The role of heads in syntax

- Heads are a central concept in linguistic theories and NLP techniques.

The role of heads in syntax

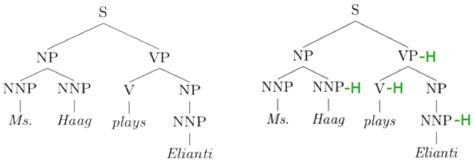
- Heads are a central concept in linguistic theories and NLP techniques.
- In **constituency structures**, the term is used to mark, for any non-terminal node, the specific daughter node that fulfills a special role.

The role of heads in syntax

- Heads are a central concept in linguistic theories and NLP techniques.
- In **constituency structures**, the term is used to mark, for any non-terminal node, the specific daughter node that fulfills a special role.
- Exactly one head per constituent.

The role of heads in syntax

- Heads are a central concept in linguistic theories and NLP techniques.
- In **constituency structures**, the term is used to mark, for any non-terminal node, the specific daughter node that fulfills a special role.
- Exactly one head per constituent.



Heads in linguistic theories

Heads in linguistic theories

Zwicky (Journal of Linguistics, 1985) lists the conditions a daughter has to fulfill in order to be the head of a construct, according to linguistic theories.

1 Is the constituent the **semantic argument**, that is, the constituent whose meaning serves as argument to some functor?
2 Is it the **determinant of concord**, that is, the constituent with which co-constituents must agree?
3 Is it the **morphosyntactic locus**, that is, the constituent which bears inflections marking syntactic relations between the whole construct and other syntactic units?
4 Is it the **subcategorizand**, that is, the constituent which is subcategorized with respect to its sisters?
5 Is it the **governor**, that is, the constituent which selects the morphological form of its sisters?
6 Is it the **distributional equivalent**, that is, the constituent whose distribution is identical to that of the whole construct?
7 Is it the **obligatory** constituent, that is, the constituent whose removal forces the whole construct to be recategorized?
8 Is it the **ruler** in dependency theory, that is, the constituent on which others depend in a dependency analysis?

Heads in NLP

- Heads are relevant for many NLP applications (parsing, MT, SRL, ...).

Heads in NLP

- Heads are relevant for many NLP applications (parsing, MT, SRL, ...).
- Heads are used because they work!

Heads in NLP

- Heads are relevant for many NLP applications (parsing, MT, SRL, ...).
- Heads are used because they work!
- Serve to have better probability distributions over the productive rules.

Heads in NLP

- Heads are relevant for many NLP applications (parsing, MT, SRL, ...).
- Heads are used because they work!
- Serve to have better probability distributions over the productive rules.
- Little attempt to have empirical (corpus based) evaluation of head assignments (only exception Chiang & Bikel 2002).

Heads in NLP

- Heads are relevant for many NLP applications (parsing, MT, SRL, ...).
- Heads are used because they work!
- Serve to have better probability distributions over the productive rules.
- Little attempt to have empirical (corpus based) evaluation of head assignments (only exception Chiang & Bikel 2002).
- Our goal is to contribute to both theory and applications, by providing algorithms for head assignments, and propose empirical evaluations.

Hand-written rules

- Predefined rules based on the labels of parent, daughters, and their positions.

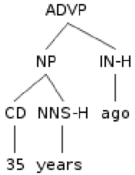
Hand-written rules

- Predefined rules based on the labels of parent, daughters, and their positions.
- Language and corpus specific.

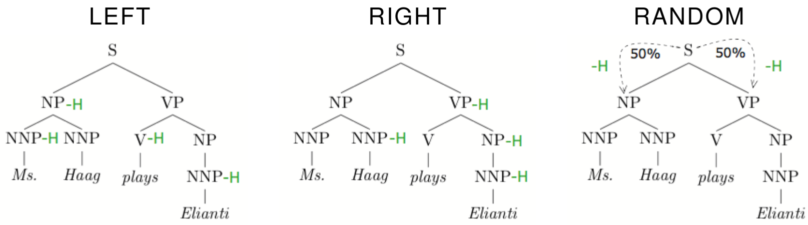
Hand-written rules

- Predefined rules based on the labels of parent, daughters, and their positions.
- Language and corpus specific.
- Magerman 95
- Collins 97
- Yamada-Matsumoto 03

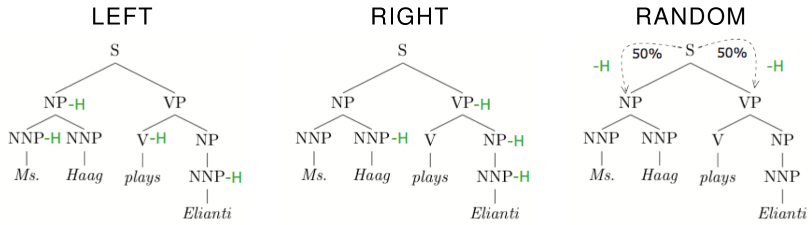
Parent Non-terminal	**Direction**	**Priority List**
...	...	...
ADVP	Right	RB RBR RBS FW ADVP TO CD JJR JJ IN NP JJS NN
NP	Right	NN NNP NNPS NNS NX POS JJR
...	...	...



Baselines

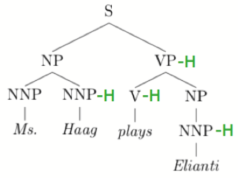

LEFT
S
NP-H
VP
NNP-H
NNP
V-H
NP
Ms.
Haag
plays
NNP-H
Elianti
RIGHT
S
NP
VP-H
NNP
NNP-H
V
NP-H
Ms.
Haag
plays
NNP-H
Elianti
RANDOM
-H
50%
S
50%
-H
NP
VP
NNP
NNP
V
NP
Ms.
Haag
plays
NNP
Elianti

Baselines

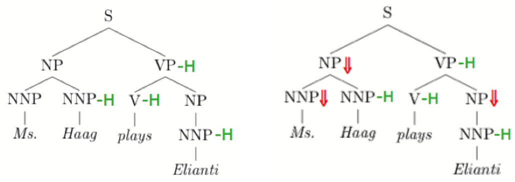


- Can we do better?

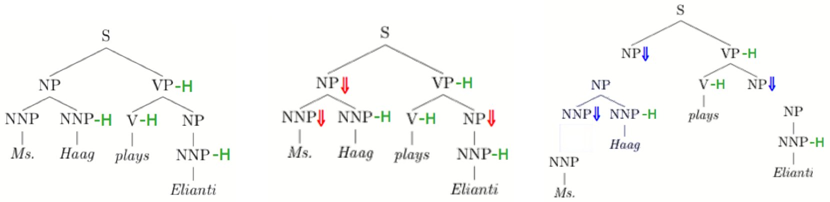
Using heads to extract (one-anchor) Lexicalized Trees



Using heads to extract (one-anchor) Lexicalized Trees

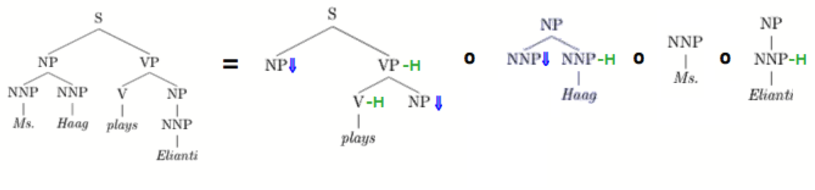


Using heads to extract (one-anchor) Lexicalized Trees



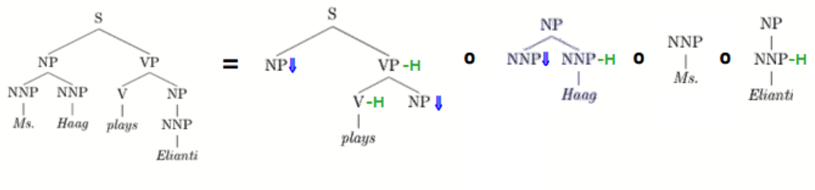
LTSGs

Lexicalized Trees + substitution operation = LTSG



LTSGs

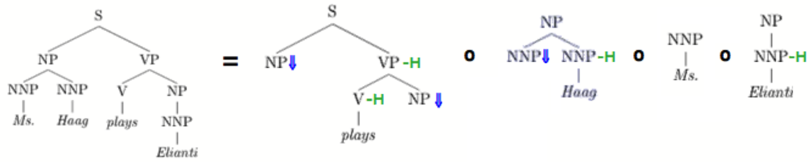
Lexicalized Trees + substitution operation = LTSG



- Corpus + Heads $\rightarrow$ LTSG

LTSGs

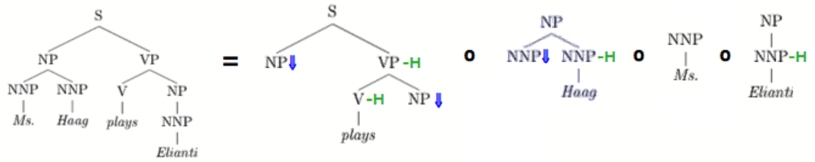
Lexicalized Trees + substitution operation = LTSG



- Corpus + Heads $\rightarrow$ LTSG
- LTSGs belong to the family of TSGs (as CFGs and DOP).

LTSGs

Lexicalized Trees + substitution operation = LTSG



- Corpus + Heads $\rightarrow$ LTSG
- LTSGs belong to the family of TSGs (as CFGs and DOP).
- As all other TSGs, LTSGs can be defined within a stochastic model.

Learning heads through LTSGs

- Given a corpus, there is a one to one mapping between head assignments and LTSGs we can extract.

Learning heads through LTSGs

- Given a corpus, there is a one to one mapping between head assignments and LTSGs we can extract.
- We define an objective function over LTSGs, based on statistical information over the lexicalized trees.

Learning heads through LTSGs

- Given a corpus, there is a one to one mapping between head assignments and LTSGs we can extract.
- We define an objective function over LTSGs, based on statistical information over the lexicalized trees.
- Choosing a head assignment = find the LTSG which optimizes the function.

Learning heads through LTSGs

- Given a corpus, there is a one to one mapping between head assignments and LTSGs we can extract.
- We define an objective function over LTSGs, based on statistical information over the lexicalized trees.
- Choosing a head assignment = find the LTSG which optimizes the function.
- **Unsupervised**: we don't learn from any given gold head assignment.

FAMILIARITY MAXIMIZATION

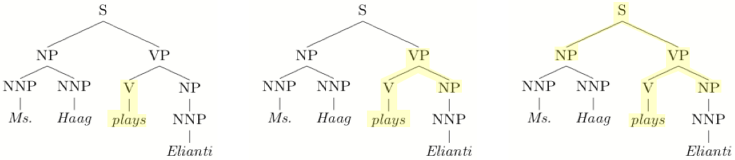
- Use elementary trees which are general enough to occur in many possible constructions.

FAMILIARITY MAXIMIZATION

- Use elementary trees which are general enough to occur in many possible constructions.
- We start by collecting bag of all lexicalized trees from the training corpus, consistent with any head annotation.

FAMILIARITY MAXIMIZATION

- Use elementary trees which are general enough to occur in many possible constructions.
- We start by collecting bag of all lexicalized trees from the training corpus, consistent with any head annotation.

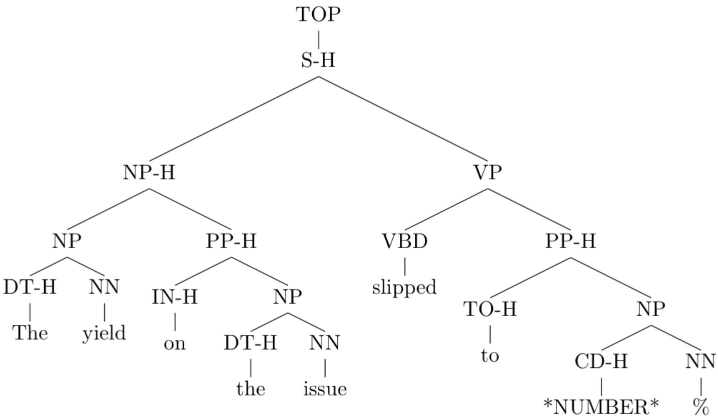


FAMILIARITY MAXIMIZATION

- We assign heads in a greedy TOP-DOWN manner: for each node we select the most frequent lexical tree rooted in it.

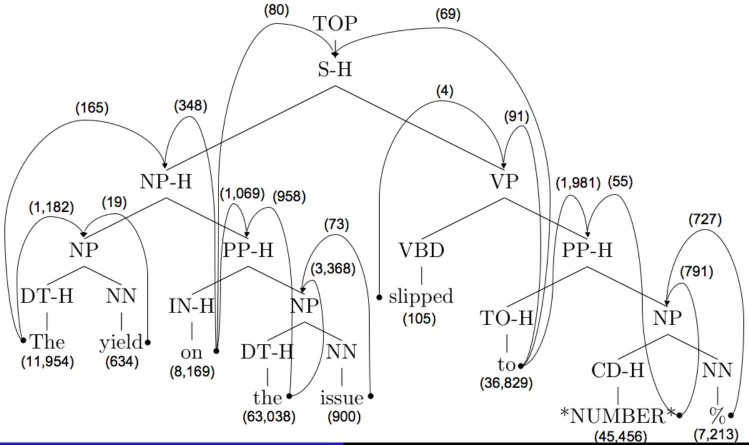
FAMILIARITY MAXIMIZATION

- We assign heads in a greedy TOP-DOWN manner: for each node we select the most frequent lexical tree rooted in it.



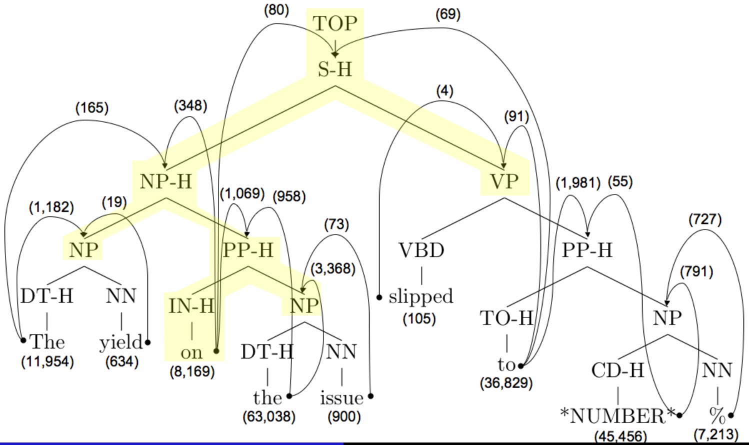
FAMILIARITY MAXIMIZATION

- We assign heads in a greedy TOP-DOWN manner: for each node we select the more frequent lexical tree rooted in it.



FAMILIARITY MAXIMIZATION

- We assign heads in a greedy TOP-DOWN manner: for each node we select the more frequent lexical tree rooted in it.



Other methods and variations

- ENTROPY MINIMIZATION: reduce the uncertainty of the structures which can be associated to each word.

Other methods and variations

- ENTROPY MINIMIZATION: reduce the uncertainty of the structures which can be associated to each word.
- EM: find the probabilistic distributions over the fragments which maximizes the likelihood of the observed data.

Other methods and variations

- ENTROPY MINIMIZATION: reduce the uncertainty of the structures which can be associated to each word.
- EM: find the probabilistic distributions over the fragments which maximizes the likelihood of the observed data.
- Variations of the algorithms: changing the distribution over the elementary trees.
    - Spine reduction (considering only the spine)
    - POStag reduction (removing words)

Evaluations

We evaluate the different head assignments in three different tasks.

- Constituency parsing (LTSG and Collins)

Evaluations

We evaluate the different head assignments in three different tasks.

- Constituency parsing (LTSG and Collins)
- Gold standard head-annotated corpus

Evaluations

We evaluate the different head assignments in three different tasks.

- Constituency parsing (LTSG and Collins)
- Gold standard head-annotated corpus
- Dependency parsing

Evaluations

We evaluate the different head assignments in three different tasks.

- Constituency parsing (LTSG and Collins) (English)
- Gold standard head-annotated corpus (English)
- Dependency parsing (English)

Evaluations

We evaluate the different head assignments in three different tasks.

- Constituency parsing (LTSG and Collins) (English)
- Gold standard head-annotated corpus (English) (German)
- Dependency parsing (English)

LTSG Parsing Results

- Corpus: Penn Wall Street Journal
- Training: sec 02-21 (sentences up to length 20)
- Test: sec 22 (sentences up to length 20)
- Parsing with our custom built LTSG parser

LTSG Parsing Results

- Corpus: Penn Wall Street Journal
- Training: sec 02-21 (sentences up to length 20)
- Test: sec 22 (sentences up to length 20)
- Parsing with our custom built LTSG parser

	LF	UF	$\lvert T \rvert$
PCFG	78.24	82.17	-

LTSG Parsing Results

- Corpus: Penn Wall Street Journal
- Training: sec 02-21 (sentences up to length 20)
- Test: sec 22 (sentences up to length 20)
- Parsing with our custom built LTSG parser

	LF	UF	$\|T\|$
PCFG	78.24	82.17	-
Magerman	**79.12**	**82.72**	**56K**
Collins97	79.05	82.71	55K
YM	79.01	82.37	56K

LTSG Parsing Results

- Corpus: Penn Wall Street Journal
- Training: sec 02-21 (sentences up to length 20)
- Test: sec 22 (sentences up to length 20)
- Parsing with our custom built LTSG parser

	LF	*UF*	$\|T\|$
PCFG	78.24	82.17	-
Magerman	**79.12**	**82.72**	**56K**
Collins97	79.05	82.71	55K
YM	79.01	82.37	56K
RANDOM	**82.89**	**85.90**	**64K**
LEFT	80.05	83.19	46K
RIGHT	70.19	75.07	51K

LTSG Parsing Results

- Corpus: Penn Wall Street Journal
- Training: sec 02-21 (sentences up to length 20)
- Test: sec 22 (sentences up to length 20)
- Parsing with our custom built LTSG parser

	LF	*UF*	$\|T\|$
PCFG	78.24	82.17	-
Magerman	**79.12**	**82.72**	**56K**
Collins97	79.05	82.71	55K
YM	79.01	82.37	56K
RANDOM	**82.89**	**85.90**	**64K**
LEFT	80.05	83.19	46K
RIGHT	70.19	75.07	51K
FAMILIARITY	**84.43**	**87.13**	**42K**

LTSG Parsing Results

- Corpus: Penn Wall Street Journal
- Training: sec 02-21 (sentences up to length 20)
- Test: sec 22 (sentences up to length 20)
- Parsing with our custom built LTSG parser

	LF	*UF*	$\|T\|$
PCFG	78.24	82.17	-
Magerman	**79.12**	**82.72**	**56K**
Collins97	79.05	82.71	55K
YM	79.01	82.37	56K
RANDOM	**82.89**	**85.90**	**64K**
LEFT	80.05	83.19	46K
RIGHT	70.19	75.07	51K
FAMILIARITY	**84.43**	**87.13**	**42K**

Collins Parser Results

- Corpus: Penn Wall Street Journal
- Training: sec 02-21 (sentences up to length 20)
- Test: sec 22 (sentences up to length 20)
- Parsing with Bikel's implementation of Collins' parser

Collins Parser Results

- Corpus: Penn Wall Street Journal
- Training: sec 02-21 (sentences up to length 20)
- Test: sec 22 (sentences up to length 20)
- Parsing with Bikel's implementation of Collins' parser

	LF	*UF*
Collins97	**86.20**	**88.35**
RANDOM	84.58	86.97
RIGHT	81.62	84.41
LEFT	81.13	83.95
FAMILIARITY-POStags	**86.27**	**88.32**

Note: the explicit annotation of heads in the training corpus interfears with some features of the parser.

Head gold evaluations - Parc700

- Number of sentences: 700
- Discarded (multiple heads) in Parc: 8.5 %

Head gold evaluations - Parc700

- Number of sentences: 700
- Discarded (multiple heads) in Parc: 8.5 %

	% correct
Yamada-Matsumoto	**85.33**
Collins97	84.50
Magerman	84.41

Head gold evaluations - Parc700

- Number of sentences: 700
- Discarded (multiple heads) in Parc: 8.5 %

	% correct
Yamada-Matsumoto	**85.33**
Collins97	84.50
Magerman	84.41
LEFT	47.63
RANDOM	44.33
RIGHT	40.70

Head gold evaluations - Parc700

- Number of sentences: 700
- Discarded (multiple heads) in Parc: 8.5 %

	% correct
Yamada-Matsumoto	**85.33**
Collins97	84.50
Magerman	84.41
LEFT	47.63
RANDOM	44.33
RIGHT	40.70
FAMILIARITY-POStags-Spine	**76.38**

Head gold evaluations - Tiger DB

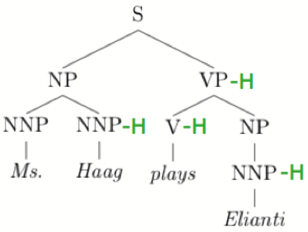
- Number of sentences: 1866
- Discarded (multiple heads) in Tiger DB: 42.9 %

	% correct
Tiger TB Head Assignment	**77.39**
RIGHT	52.59
RANDOM	38.66
LEFT	18.64
FAMILIARITY-POStags-Spine	**68.88**

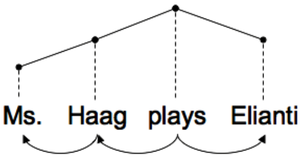
Constituency structure and Dependency structure

Heads can be seen as a bridge to convert constituency structures to dependency structures.

Constituency Structure



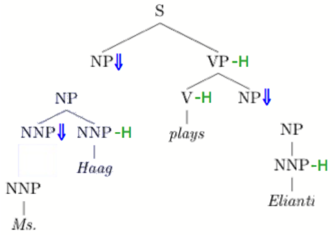
Dependency Structure



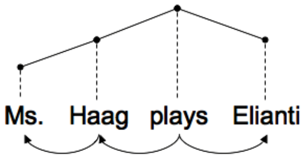
Constituency structure and Dependency structure

Heads can be seen as a bridge to convert constituency structures to dependency structures.

Constituency Structure



Dependency Structure



Dependency Parsing

- Corpus: Penn Wall Street Journal corpus
- Training: sec 02-11 unlab. (sentences up to length 20)
- Test: sec 22 unlab. (sentences up to length 20)
- MST (McDonald et al.) dependency parser
- Similar result with MALT (Nivre et al.)

	Self *UAS*	**Collins97** *UAS*
Collins97	**91.0**	100.0
Yamada-Matsumoto	90.5	86.4
Magerman	89.7	79.2
LEFT	**91.6**	23.2
RIGHT	90.0	25.7
RANDOM	20.7	22.3
FAMILIARITY-POStags-Spine	**83.9**	53.2

Conclusions

- Variations of FAMILIARITY algorithm do well in the three tasks.

Conclusions

- Variations of FAMILIARITY algorithm do well in the three tasks.
- There is no single assignment which works best in all the evaluations.

Conclusions

- Variations of FAMILIARITY algorithm do well in the three tasks.
- There is no single assignment which works best in all the evaluations.
- Evaluations of different head assignments in real NLP applications are desirable.

Conclusions

- Variations of FAMILIARITY algorithm do well in the three tasks.
- There is no single assignment which works best in all the evaluations.
- Evaluations of different head assignments in real NLP applications are desirable.
- **Head assignments as a new task in NLP!**

Thank you!

Extra material at:

`http://staff.science.uva.nl/~fsangati`

`{f.sangati,zuidema}@uva.nl`

Stochastic LTSGs

Lexicalized Trees + substitution operation = LTSG

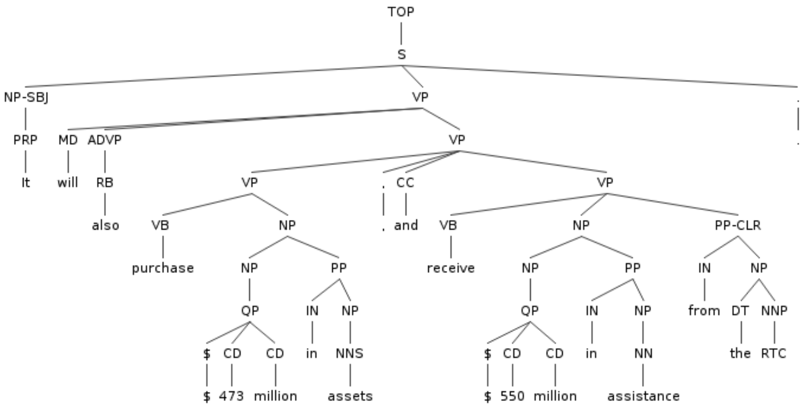
- Corpus + Heads $\rightarrow$ LTSG
- LTSGs belong to the family of TSGs (as CFGs and DOP).
- As all other TSGs, LTSGs can be defined within a stochastic model.

$$F(\tau) = \frac{f(\tau)}{\sum_{\tau' : r(\tau') = r(\tau)} f(\tau')}$$

$$P(d) = \prod_{\tau_i \in d} F(\tau_i)$$

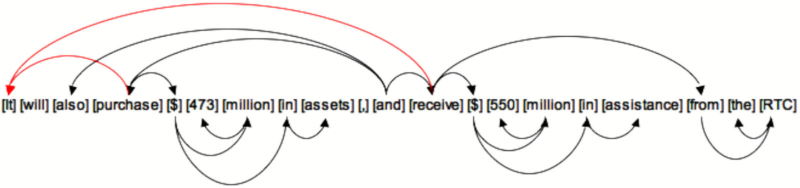
$$P(t) = \sum_{d_j \in \delta(t)} P(d_j) = \sum_{d_j \in \delta(t)} \prod_{\tau_i \in d} F(\tau_i)$$

Parc700 Original



Parc700 Dependecy Structure

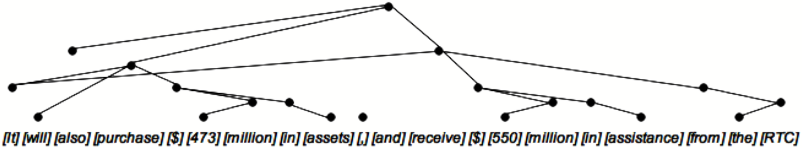
num($~12, pl)
number($~12, million~21)
pers($~12, 3)
adjunct_type(in~15, nominal)
obj(in~15, asset~16)
ptype(in~15, semantic)
num(asset~16, pl)
pers(asset~16, 3)
adjunct(million~21, 473~23)
number_type(million~21, cardinal)
number_type(473~23, cardinal)
adjunct($~25, in~31)
num($~25, pl)
number($~25, million~34)
vtype(purchase~9, main)
mood(receive~10, indicative)
obj(receive~10, $~25)
obl(receive~10, from~26)
subj(receive~10, pro~11)
tense(receive~10, fut)
vtype(receive~10, main)
case(pro~11, nom)
gend_sem(pro~11, nonhuman)
num(pro~11, sg)
pers(pro~11, 3)
pron_form(pro~11, it)
pron_type(pro~11, pers)
adjunct($~12, in~15)
adjunct(coord~0, also~8)
conj(coord~0, purchase~9)
conj(coord~0, receive~10)
coord_form(coord~0, and)
coord_level(coord~0, VPauxcoord)
stmt_type(coord~0, declarative)
adegree(also~8, positive)
adv_type(also~8, sadv)
mood(purchase~9, indicative)
obj(purchase~9, $~12)
subj(purchase~9, pro~11)
tense(purchase~9, fut)
pers($~25, 3)
obj(from~26, RTC~37)
ptype(from~26, semantic)
adjunct_type(in~31, nominal)
obj(in~31, assistance~32)
ptype(in~31, semantic)
num(assistance~32, sg)
pers(assistance~32, 3)
adjunct(million~34, 550~36)
number_type(million~34, cardinal)
number_type(550~36, cardinal)
det_form(RTC~37, the)
det_type(RTC~37, def)
num(RTC~37, sg)
pers(RTC~37, 3)
proper(RTC~37, misc))



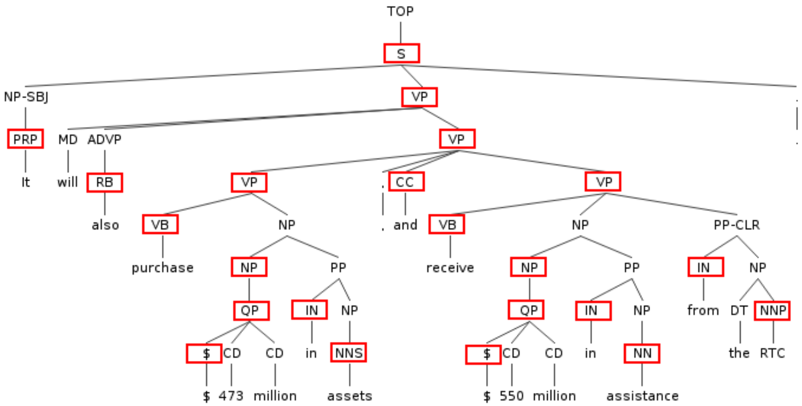
Parc700 Dependency Structure 2 (Tomas By)

```
word(2, 0, it, [index-'11',pron_type-pers,pron_form-it,pers-'3',num-sg,gend_sem-nonhuman,case-nom]).
word(2, 1, will, []).
word(2, 2, also, [index-'8',adv_type-sadv,adegree-positive]).
word(2, 3, purchase, [index-'9',vtype-main,tense-fut,mood-indicative]).
word(2, 4, $, [index-'12',pers-'3',num-pl]).
word(2, 5, '473', [index-'23',number_type-cardinal]).
word(2, 6, million, [index-'21',number_type-cardinal]).
word(2, 7, in, [index-'15',ptype-semantic,adjunct_type-nominal]).
word(2, 8, assets, [index-'16',pers-'3',num-pl]).
word(2, 9, ',', []).
word(2, 10, and, [index-'0',stmt_type-declarative,coord_level-'VPauxcoord',coord_form-and]).
word(2, 11, receive, [index-'10',vtype-main,tense-fut,mood-indicative]).
word(2, 12, $, [index-'25',pers-'3',num-pl]).
word(2, 13, '550', [index-'36',number_type-cardinal]).
word(2, 14, million, [index-'34',number_type-cardinal]).
word(2, 15, in, [index-'31',ptype-semantic,adjunct_type-nominal]).
word(2, 16, assistance, [index-'32',pers-'3',num-sg]).
word(2, 17, from, [index-'26',ptype-semantic]).
word(2, 18, the, []).
word(2, 19, 'RTC', [index-'37',proper-misc,pers-'3',num-sg,det_type-def,det_form-the]).
```

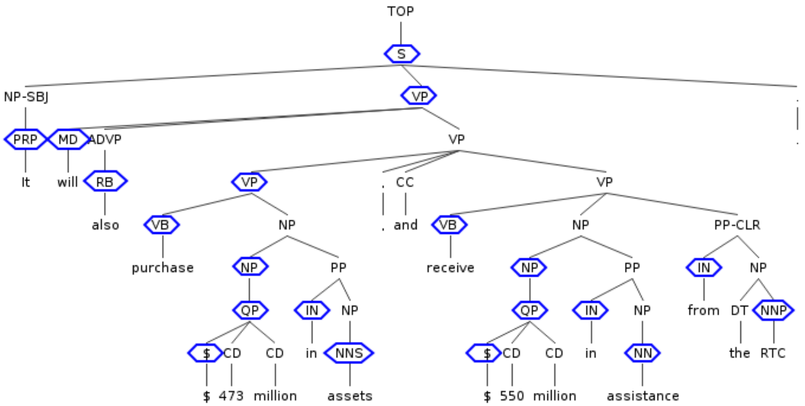
```
dependency(2, w(18), [det_form], w(19)).
dependency(2, w(1), [tense], w(3)).
dependency(2, w(2), [adjunct], w(10)).
dependency(2, w(3), [conj], w(10)).
dependency(2, w(11), [conj], w(10)).
dependency(2, w(0), [subj], w(3)).
dependency(2, w(4), [obj], w(3)).
dependency(2, w(0), [subj], w(11)).
dependency(2, w(12), [obj], w(11)).
dependency(2, w(17), [obl], w(11)).
dependency(2, w(7), [adjunct], w(4)).
dependency(2, w(6), [number], w(4)).
dependency(2, w(8), [obj], w(7)).
dependency(2, w(5), [adjunct], w(6)).
dependency(2, w(15), [adjunct], w(12)).
dependency(2, w(14), [number], w(12)).
dependency(2, w(19), [obj], w(17)).
dependency(2, w(16), [obj], w(15)).
dependency(2, w(13), [adjunct], w(14)).
```



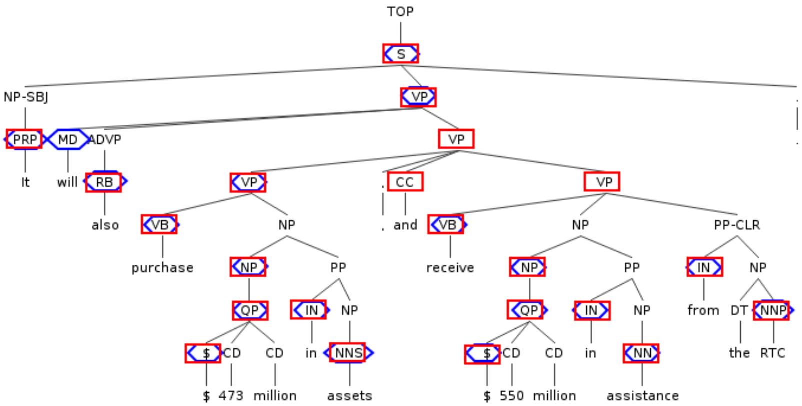
Parc700 Gold



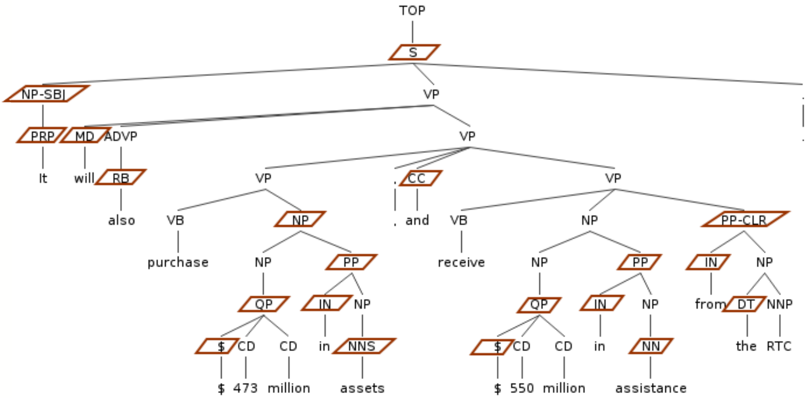
Parc700 Collins97



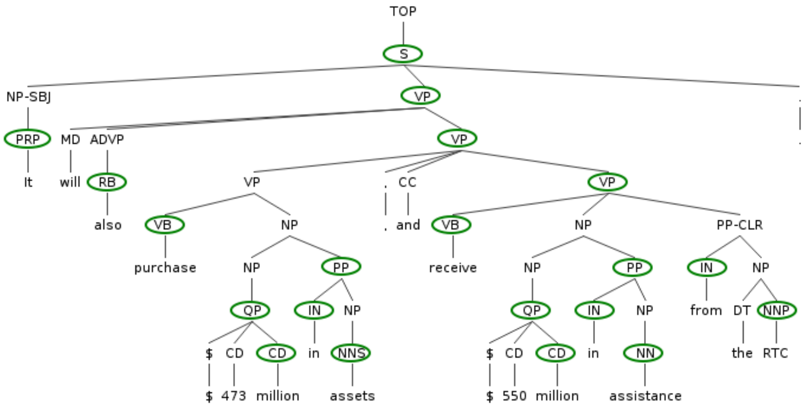
Parc700 Gold VS Collins97



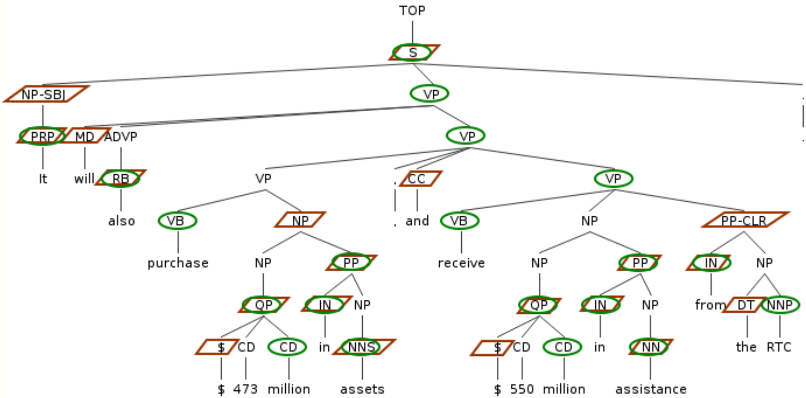
Parc700 Familiarity



Parc700 Familiarity POStag Spine



Parc700 Familiarity VS Familiarity POStag Spine



Parc700 Gold VS Familiarity POStag Spine