The SUBTLE NL Parsing Pipeline: A Complete Parser for English

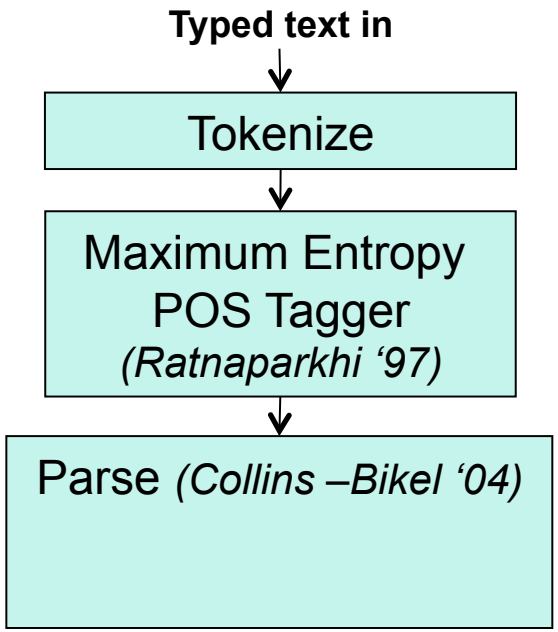
Mitch Marcus

University of Pennsylvania

PICTURE OF ANALYSIS PIPELINE

- Tokenize
- Maximum Entropy POS tagger MXPOST Ratnaparkhi
- Core Parser Collins Generative Prob. CFG
- Functional Tag Replacement – Kulick
- Null Element Placement – Ryan Gabbard PhD Dec 2010 (just submitted)

NLP Analysis Pipeline



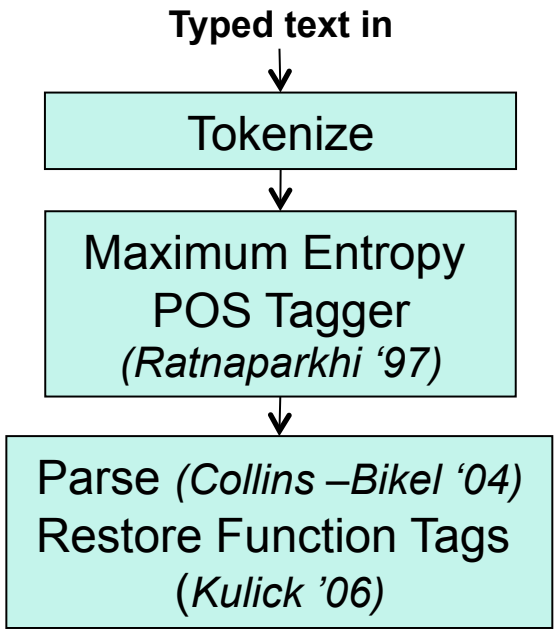
Junior, defuse any bombs immediately when going into a room.

Junior_NNP ,_, defuse_VBP any_DT bombs_NNS immediately_RB when_WRB going_VBG into_IN a_DT room_NN ._.

```
( (S  (NP        (NNP Junior))
      (, ,)

      (VP (VB defuse)
         (NP (DT any)
             (NNS bombs))
         (ADVP       (RB immediately))
         (SBAR       (WHADVP    (WRB when))
                  (S
                     (VP (VBG going)
                        (PP       (IN into)
                              (NP (DT a)
                                 (NN room)))
                                          )))))
      (. .)))
```

NLP Analysis Pipeline

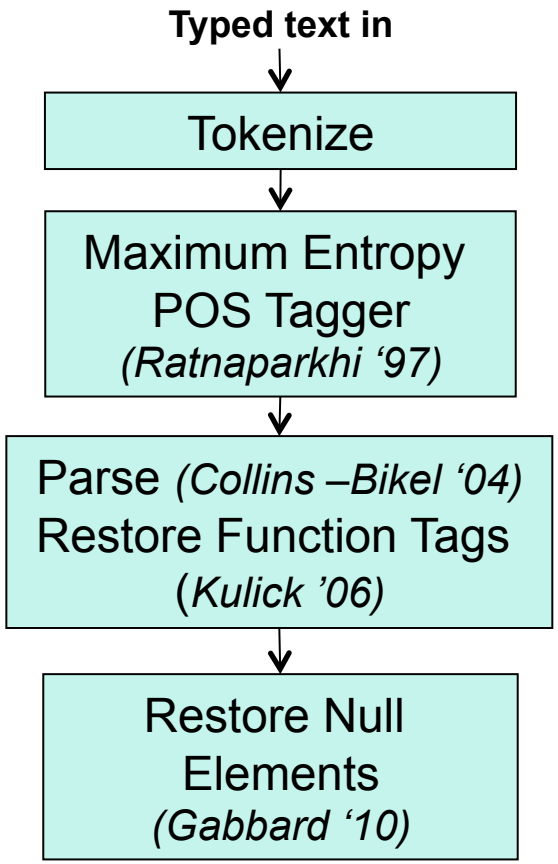


Junior, defuse any bombs immediately when going into a room.

Junior_NNP ,_, defuse_VBP any_DT bombs_NNS immediately_RB when_WRB going_VBG into_IN a_DT room_NN ._.

```
( (S (NP (NNP Junior))
     (, ,)
     (VP (VB defuse)
       (NP (DT any)
           (NNS bombs))
       (ADVP (RB immediately))
       (SBAR (WHADVP (WRB when))
             (S
               (VP (VBG going)
                 (PP (IN into)
                   (NP (DT a)
                     (NN room)))
                                  )))))
     (. .)))
```

NLP Analysis Pipeline



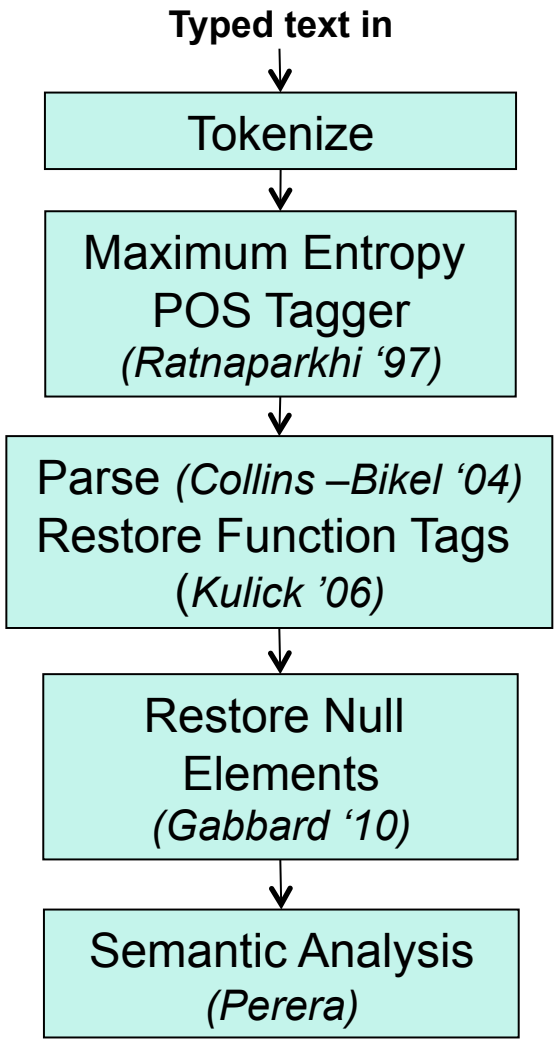
Junior, defuse any bombs immediately when going into a room.

Junior_NNP ,_, defuse_VBP any_DT bombs_NNS immediately_RB when_WRB going_VBG into_IN a_DT room_NN ._.

```
( (S (NP-VOC (NNP Junior))
     (, ,)

     (VP (VB defuse)
        (NP (DT any)
            (NNS bombs))
        (ADVP-TMP (RB immediately))
        (SBAR-TMP (WHADVP (WRB when))
                  (S
                     (VP (VBG going)
                        (PP-DIR (IN into)
                                (NP (DT a)
                                    (NN room)))
                                                 )))))
     (. .)))
```

NLP Analysis Pipeline



Junior, defuse any bombs immediately when going into a room.

Junior_NNP ,_, defuse_VBP any_DT bombs_NNS immediately_RB when_WRB going_VBG into_IN a_DT room_NN ._.

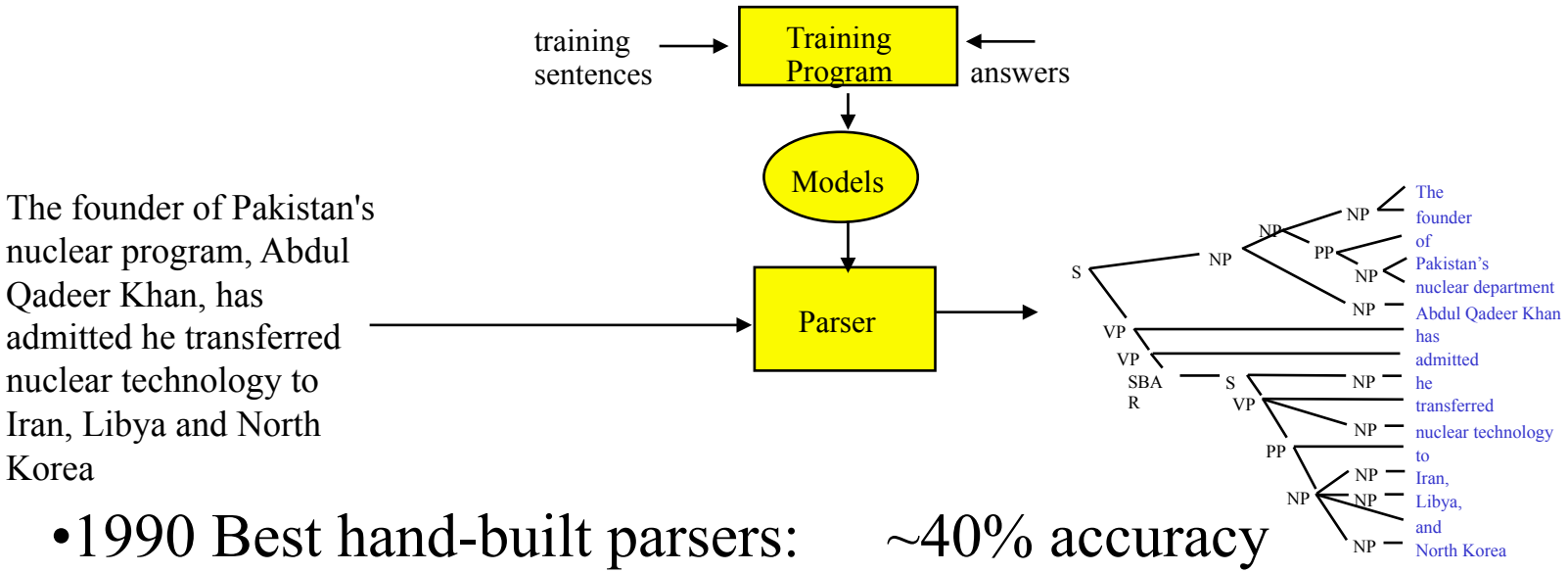
```
( (S (NP-VOC (NNP Junior))
     (, ,)
     (NP-SBJ-0 (-NONE- *))
     (VP (VB defuse)
        (NP (DT any)
            (NNS bombs))
        (ADVP-TMP (RB immediately))
        (SBAR-TMP (WHADVP-1 (WRB when))
                  (S (NP-SBJ-0 (-NONE- *))
                     (VP (VBG going)
                        (PP-DIR (IN into)
                                (NP (DT a)
                                    (NN room)))
                        (ADVP-1 (-NONE- *T*))))))
     (. .)))
```


1995: A breakthrough in parsing

10^6 words of Penn Treebank Annotation

+ Machine Learning = Robust Parsers

(Magerman '95)



- 1990 Best hand-built parsers: ~40% accuracy
- 1995+ Statistical parsers: >90% accuracy

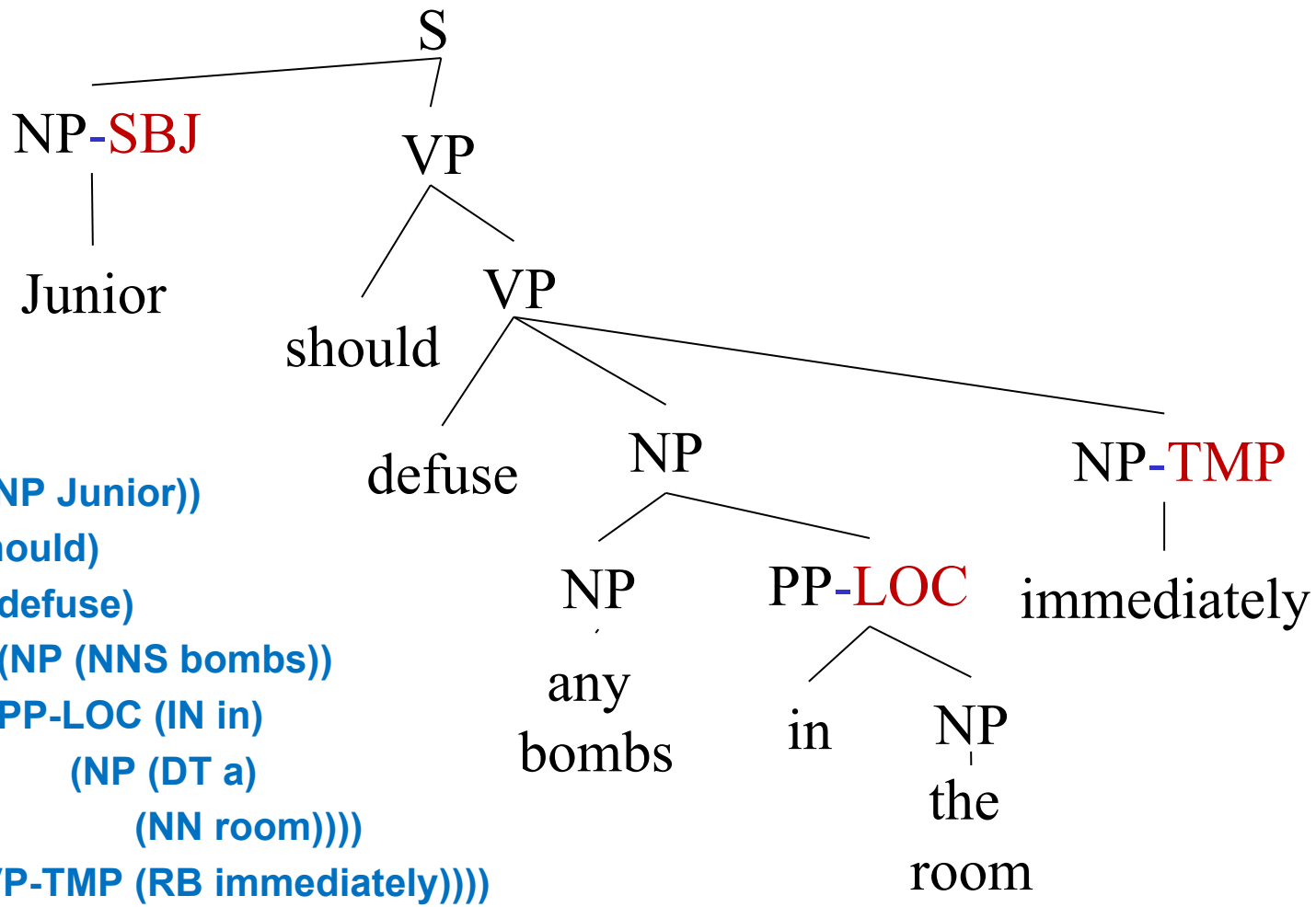
(both on short sentences)

Generative Parsing Results (<40 w)

Parser	F_1
Magerman 95	84.7
Collins 96	86.0
Charniak 97	87.4
Collins 99 (Bikel 04)	88.6
Liang 08 (extending Charniak 05)	91.7

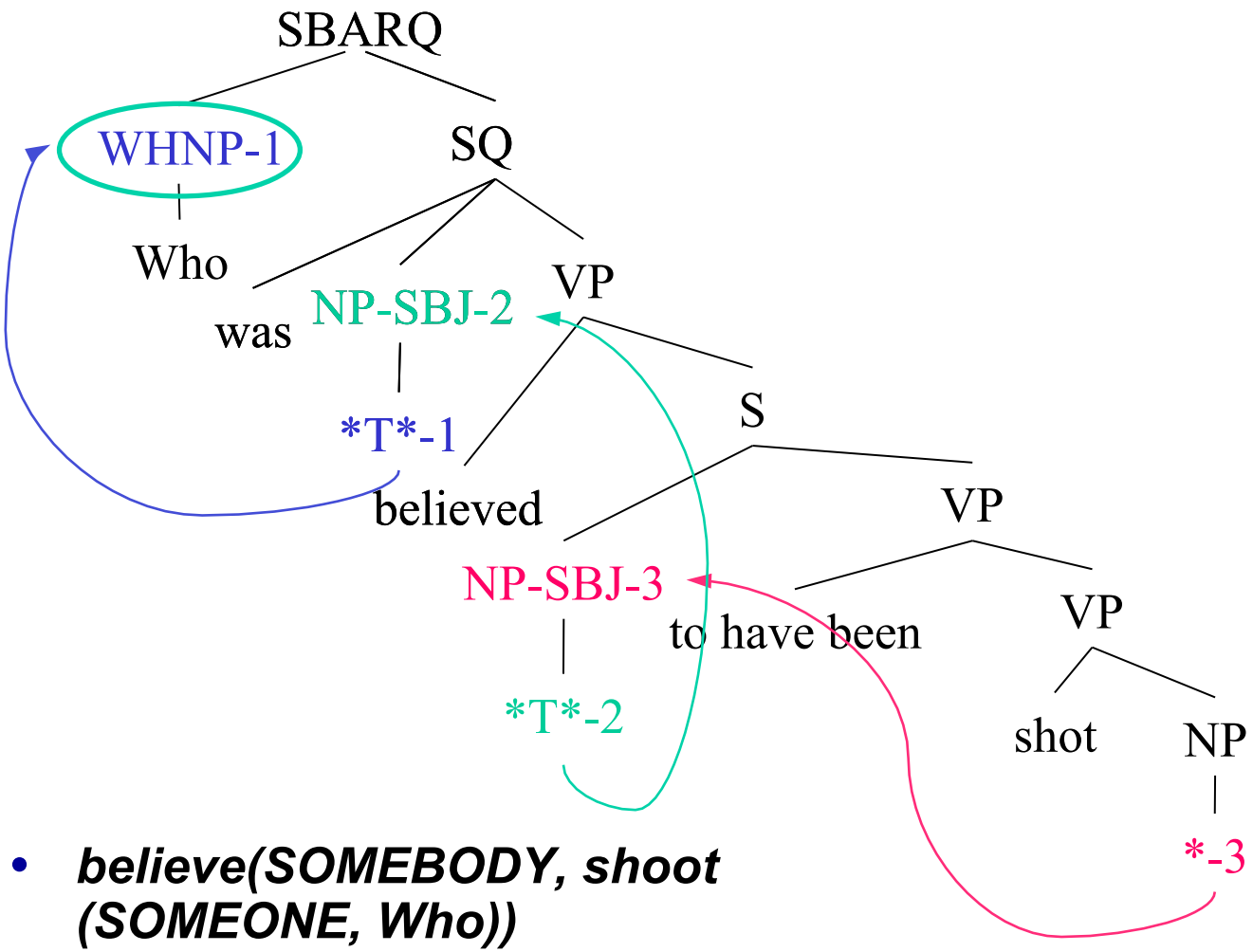
Meaningless Progress…..

Treebank Tree with Function Tags



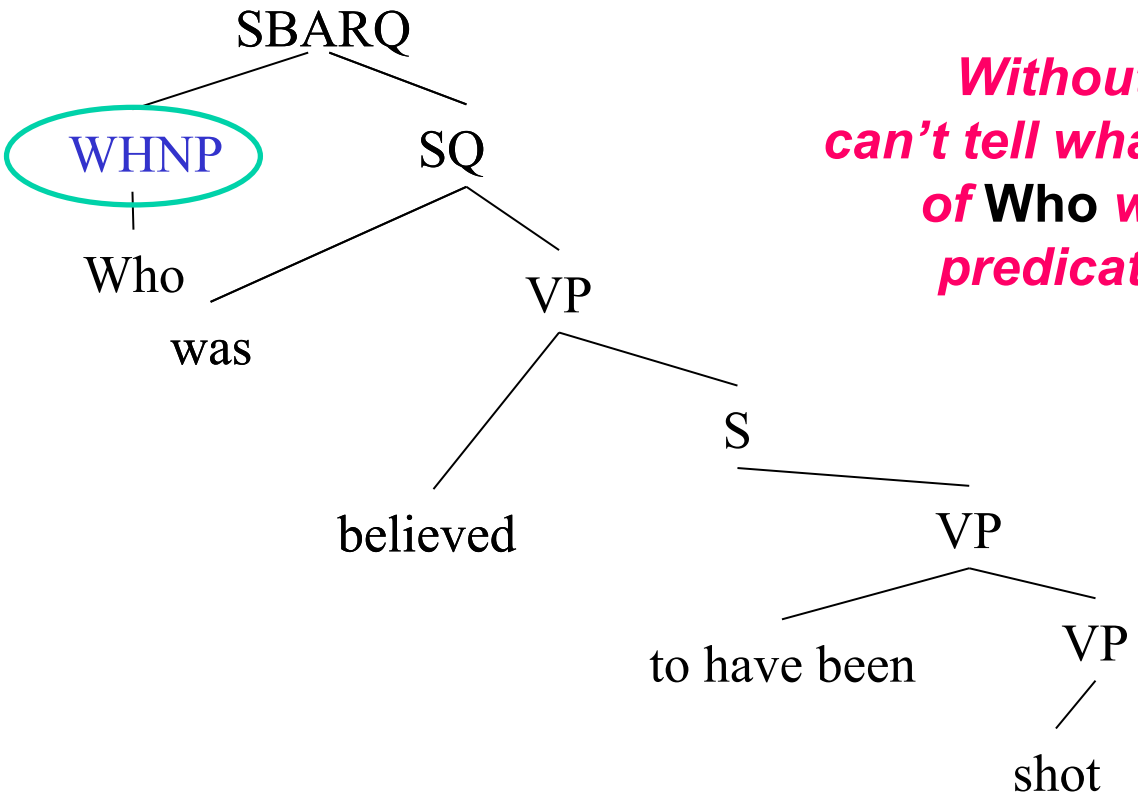

```
(S  (NP-SBJ (NNP Junior))
    (VP (MD should)
      (VP (VB defuse)
        (NP  (NP (NNS bombs))
          (PP-LOC (IN in)
            (NP (DT a)
              (NN room))))
        (ADVP-TMP (RB immediately)))))
```

Treebank Tree with Null Elements



- ***believe(SOMEBODY, shoot (SOMEONE, Who))***

A "Perfect" Parseval Analysis..



Without null element, can't tell what the semantic role of* Who *was, or even the predication it's part of.

- ***believe(SOMEBODY, shoot (SOMEONE, Who))***

A Complete Parsing System

- **Ryan Gabbard, Seth Kulick, and Mitch Marcus. Fully Parsing the Penn Treebank. HLT/NAACL 2006**
- **Ryan Gabbard, Null Element Restoration, PhD Dissertation, Nov 2010**

Step 1: Recovering Function Tags

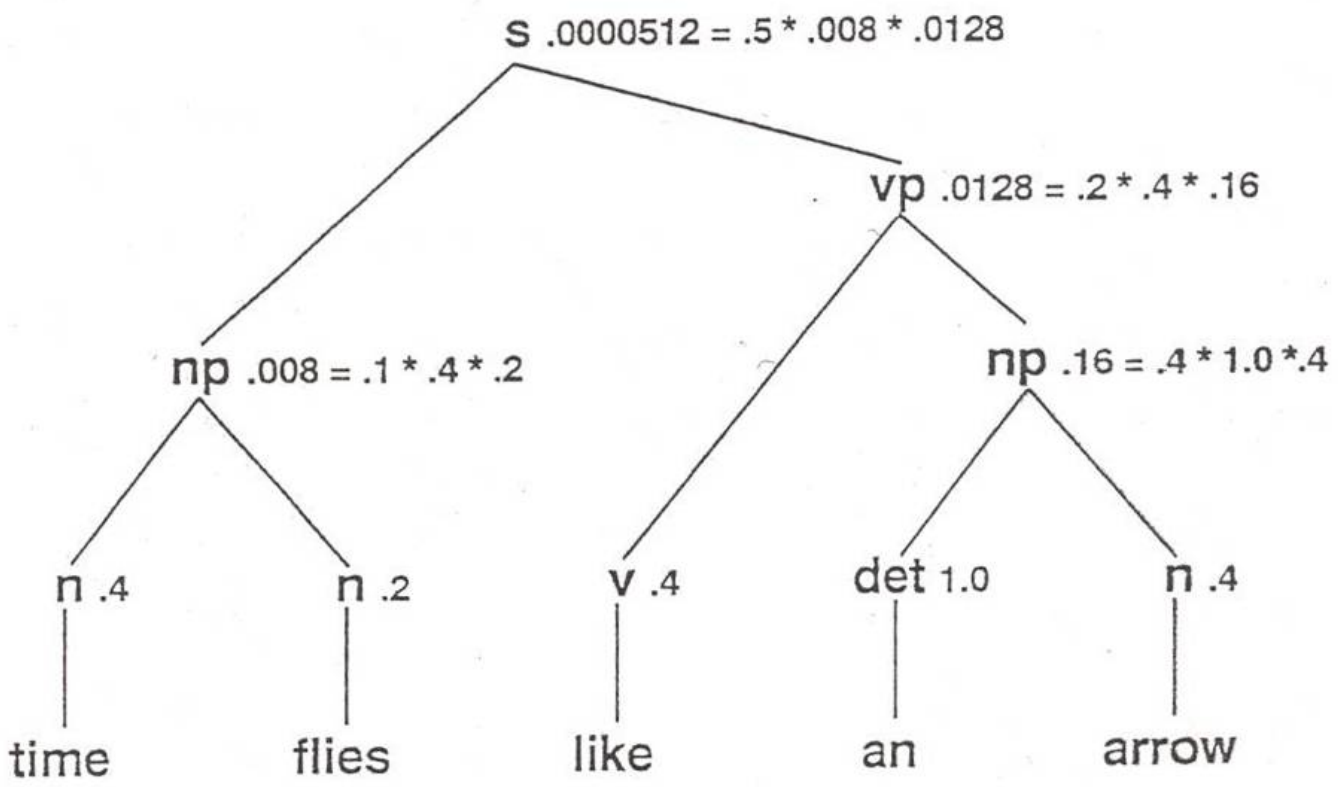
Treebank II Function Tag Set

Syntactic		*Semantic*		*Miscellaneous*	
SBJ	Subject	TMP	Temporal	CLF	It-cleft
LGS	Logical Subject	LOC	Location	HLN	Headline
PRD	Predicate	DIR	Direction	TTL	Title
DTV	Dative	EXT	Extent	*Topicalization*	
PUT	Locative of "put"	BNF	Benefactive	TPC	Topic
VOC	Vocative	MNR	Manner		
		PRP	Purpose		
		NOM	Nominal	*CLR*	
		ADV	Non-specific adverbial	CLR	Closely-Related

A Sample PCFG

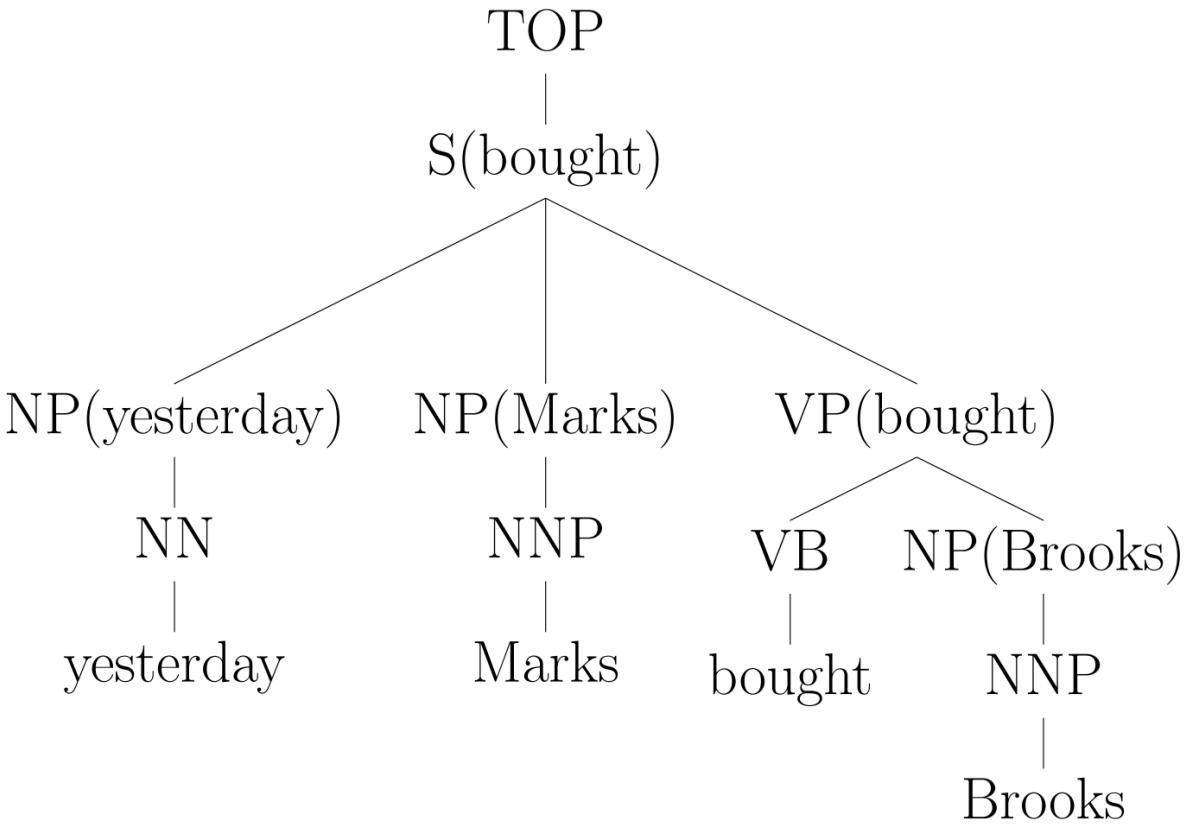
1. s $\xrightarrow{.5}$ np vp
2. s $\xrightarrow{.25}$ s pp
3. s $\xrightarrow{.15}$ v np
4. s $\xrightarrow{.10}$ v pp
5. vp $\xrightarrow{.2}$ v np
6. vp $\xrightarrow{.4}$ v pp
7. vp $\xrightarrow{.4}$ v
8. np $\xrightarrow{.1}$ n n
9. np $\xrightarrow{.4}$ det n
10. np $\xrightarrow{.3}$ np pp
11. np $\xrightarrow{.2}$ n
12. n $\xrightarrow{.4}$ arrow
13. n $\xrightarrow{.2}$ flies
14. n $\xrightarrow{.4}$ time
15. v $\xrightarrow{.4}$ flies
16. v $\xrightarrow{.4}$ like
17. v $\xrightarrow{.2}$ time
18. p $\xrightarrow{1.0}$ like
19. det $\xrightarrow{1.0}$ a
20. pp $\xrightarrow{1.0}$ p np

An Example Derivation



This example adapted from Jelinek et. al. 1991.

The Key Trick in '95: Lexicalizing the CFG



New problem: Estimating the probability of
S(bought)→NP(yesterday) NP(Marks) VP(bought)

Solving the Sparse Data Problem with Independence Assumptions

Instead of

S(bought)→NP(yesterday) NP(Marks) VP(bought)

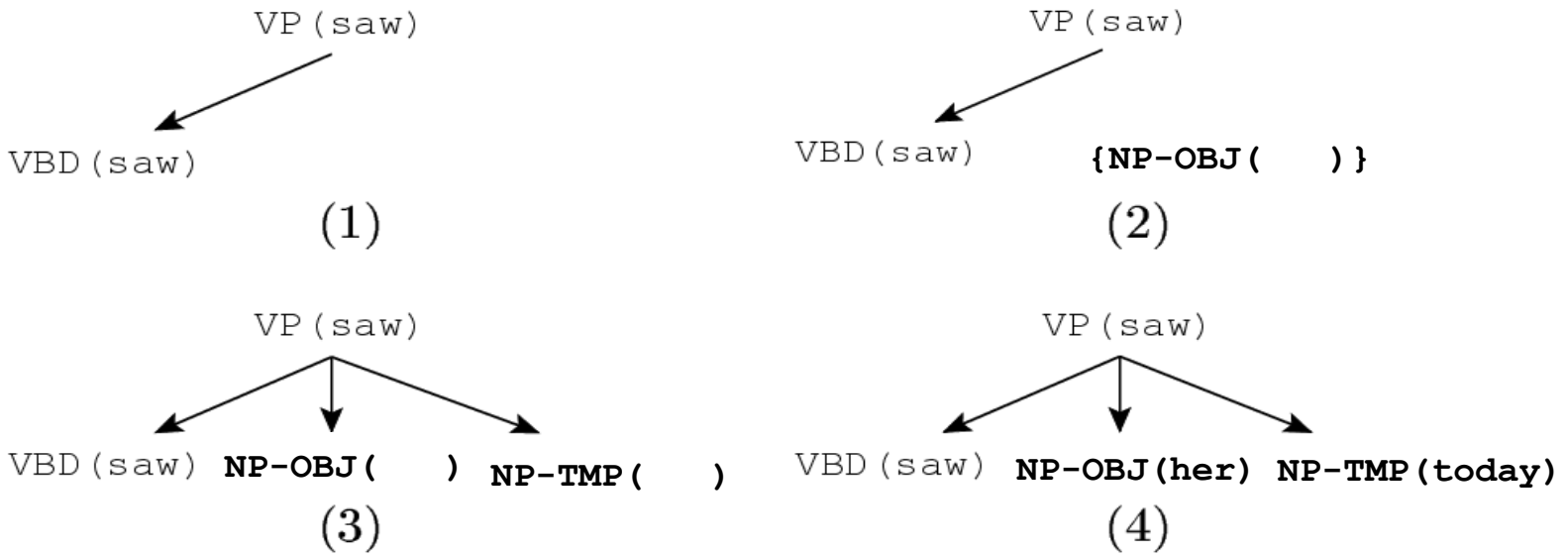
1. **Generate Head, given lexicalized parent:**
   S(bought) → …VP(bought) ….

2. **Generate other children given parent & head:**
   S(bought) → NP( ) NP ( ) VP(bought)

3. **Lexicalize other children, given parent, head & node type:**
   S(bought) → NP(yesterday)NP(Marks) VP(bought)

To augment the parser output with function tags

1. Find the code in Collins Model II that removes function tags
2. Remove it!



(From Collins & Koo '05)

Results for Function Tag Groups & Overall

	Overall w/ CLR	Overall w/o CLR
Collins 2 +FTags	*88.95*	*90.78*
Blaheta, 2003		88.28
Jijkoun & de Rijke, 2004	88.50	

- With no drop in basic parser F score!

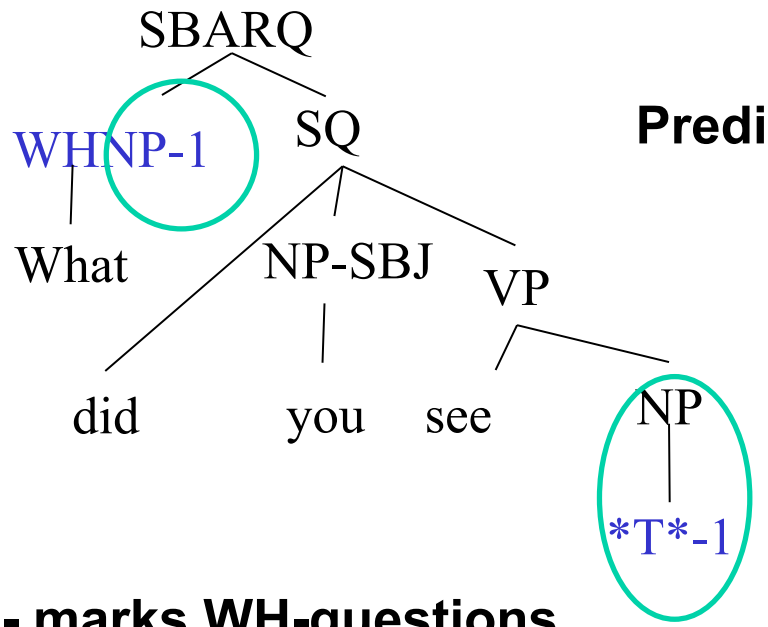
Results for Function Tag Groups & Overall

	Overall w/ CLR	Overall w/o CLR	Syn 56%	Sem 36%
Collins 2 +FTags	*88.95*	*90.78*	95.76	84.56
Blaheta, 2003		88.28		
Jijkoun & de Rijke, 2004	88.50			
Musillo & Merlo, Oct 2005			*96.5*	*85.6*

- With no drop in basic parser F score!

Step 2: Recovering Null Elements

Null Elements for "Wh-movement"



Predicate argument structure:
see(your,what)

- ***T* - marks WH-questions**
- **Null elements are co-indexed using integer tags.**
- **To recover Pred-arg structure:**
  - Replace null element with co-indexed lexical material

- ***Similar approaches for other grammatical phenomenon***

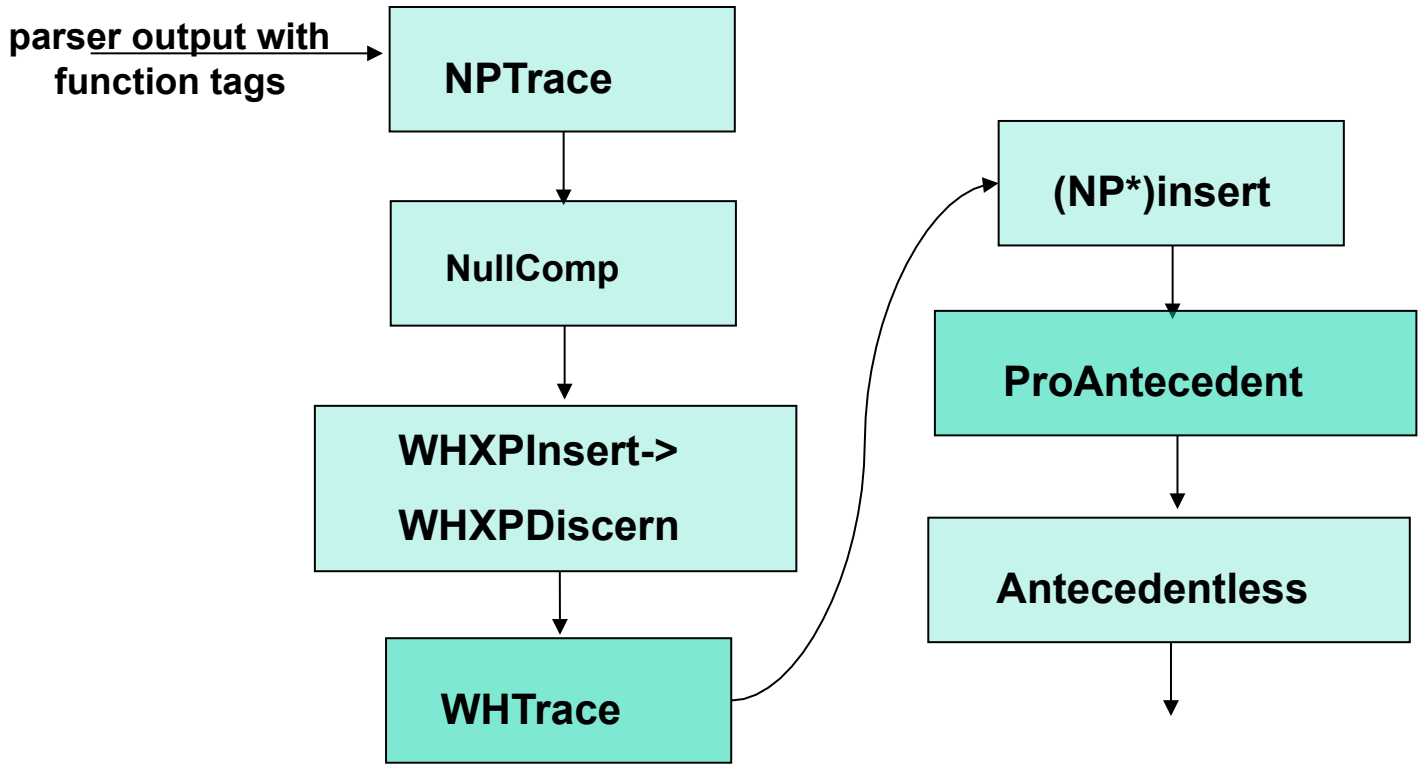
Previous Work Forward

- **Integrated with Parsing (Collins Model 3, Dienes&Dubey 03)**
- **Postprocessing Parser Output**
  - **Hand Built Rule-based (**Campbell 04)
  - **Statistical Machine Learning Approaches**
    (Johnson 02, Levy&Manning 04, Jijkoun&de Rijke 04)

- **Our approach combines:**
  - **Campbell's Problem Decomposition**
  - **Machine Learning**
- Into:
  - **A pipeline of linear classifiers**

The Null Element Replacement Algorithm

- Pipeline & rich linguistic features á la Campbell
- Pipeline of linear classifiers (McCallum's Mallet)



Aggregate results over all recovered categories

System	Precision	Recall	F1
D&D	**78.50**	68.08	72.92
Pres	74.70	**74.62**	**74.66**

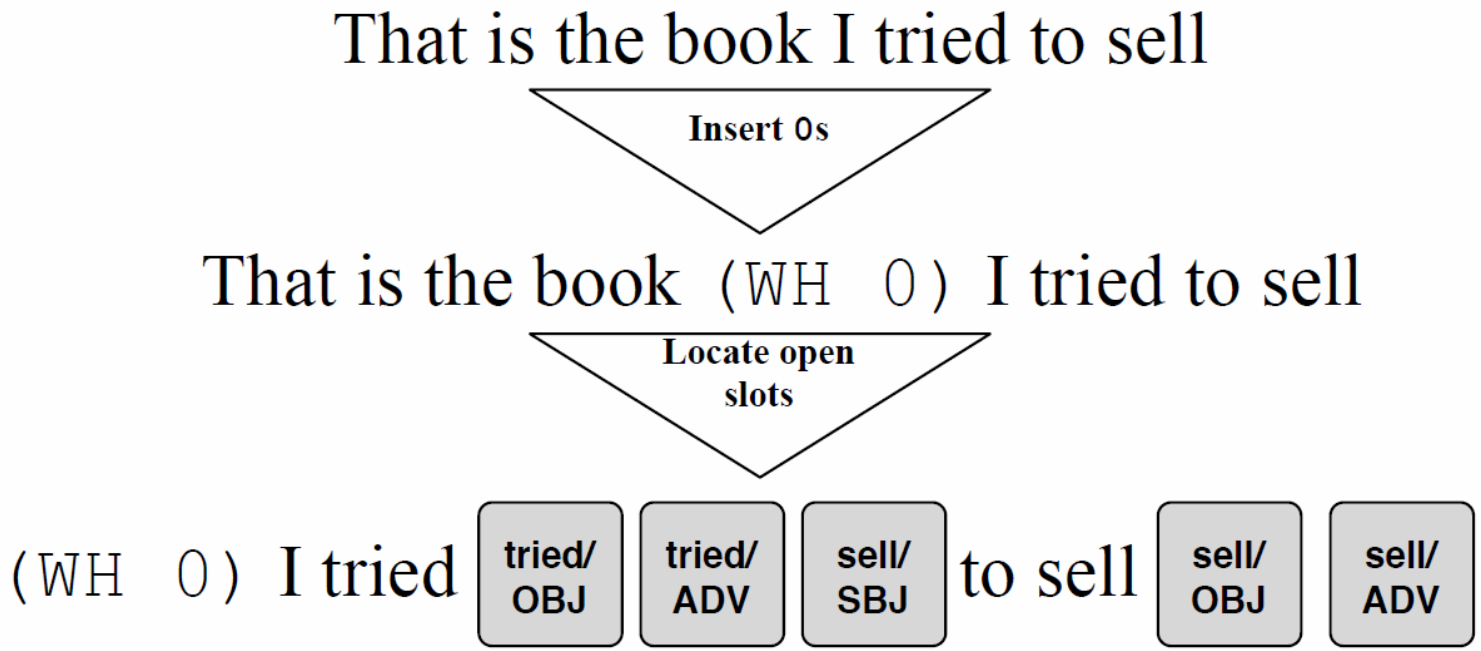
- **Automatically parsed trees from section 23**

Present vs. Campbell's Rule Based System

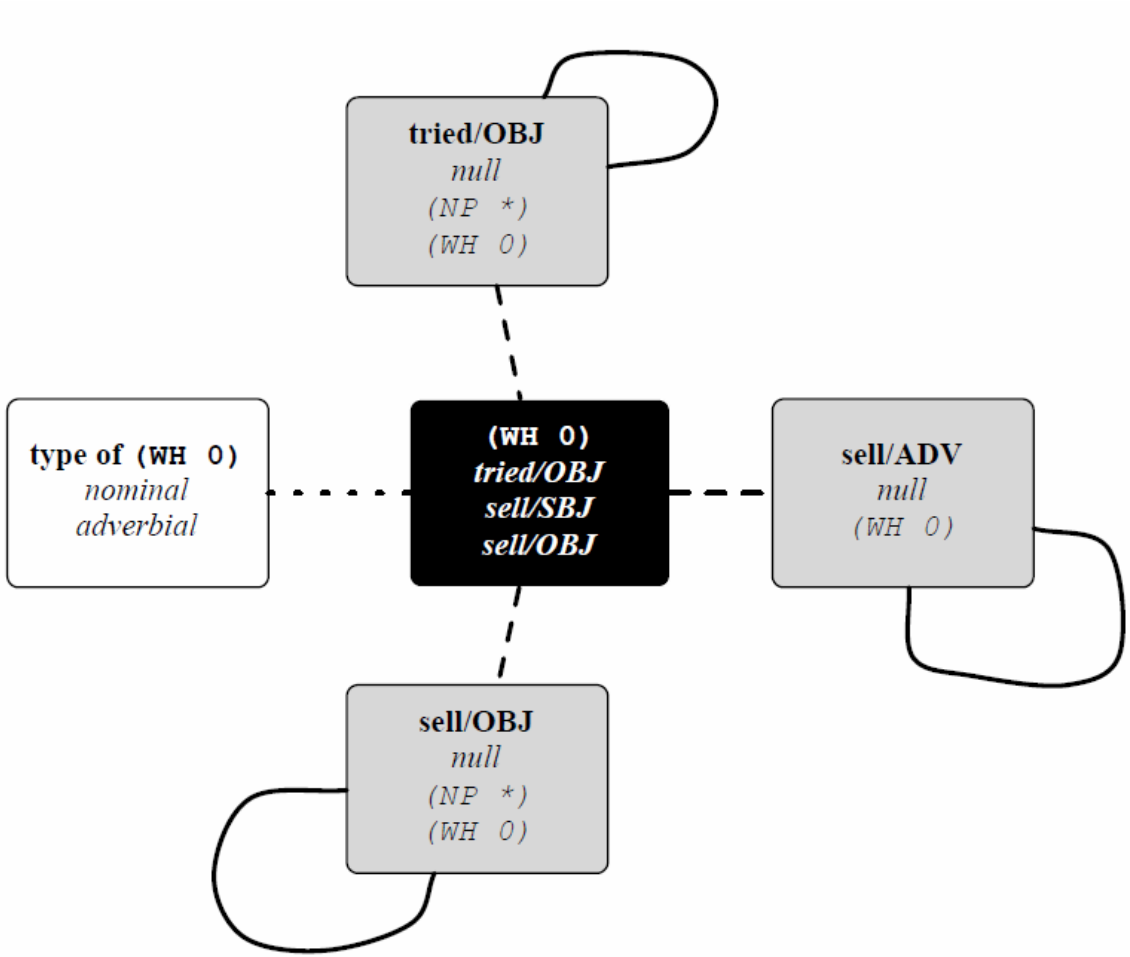
- **Test on gold--standard trees from section 23 using Campbell's metric**

Category	Present	Campbell
NP *	**88.8**	86.9
NP *T*	**96.3**	96.0
ADVP *T*	**82.2**	79.9
0	**99.8**	98.5

Wh – placement with factor graphs



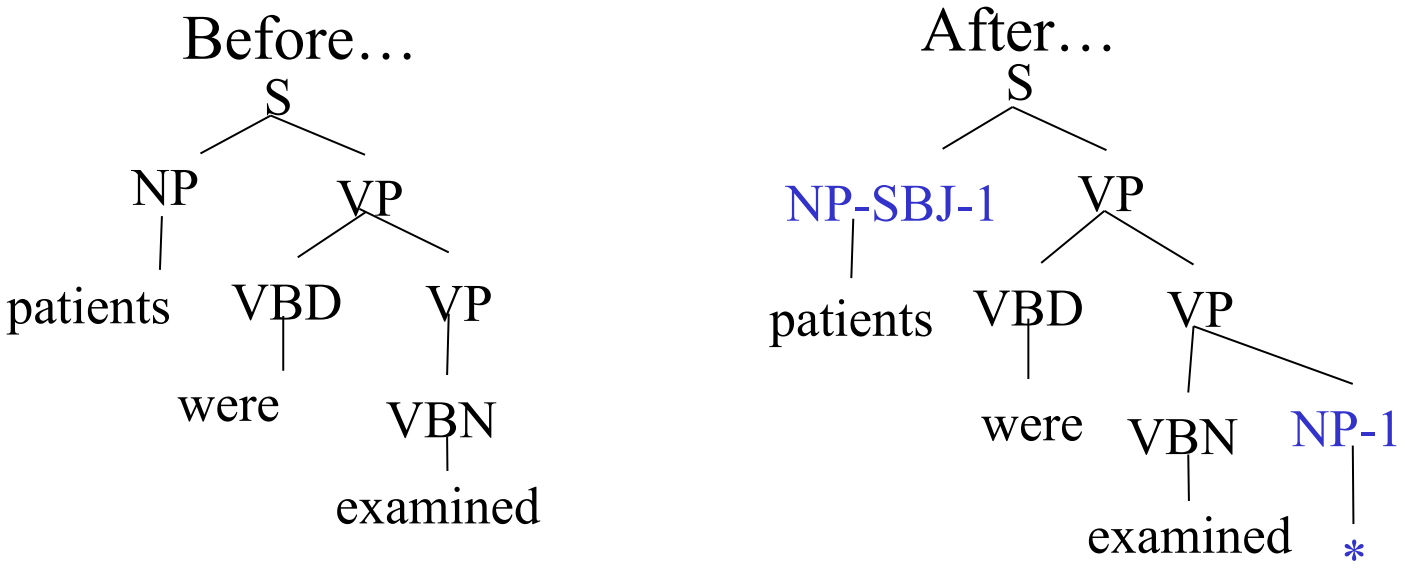
Wh – placement with factor graphs



Results: two systems for Wh-placement

Type	Gold		Parsed	
	Old	New	Old	New
NP *T*	92.8	**96.5**	85.9	**87.9**
ADVP *T*	79.2	**86.7**	77.7	**79.8**
NP *	78.6	**82.7**	**72.2**	71.6
*NP * (na)*	95.8	**96.6**	**88.3**	88.0
WHNP 0	**92.0**	90.2	**61.5**	59.6
WHADVP 0	**71.0**	66.7	**68.9**	61.8

Summary: Full Parsing!



- **20 Function Tags – Mostly syntactic or semantic**
  - **Syntactic Tags: 95% recovery accuracy**
  - **Semantic Tags: 84% recovery accuracy**
- **Traces**
  - **Passivization, control, raising: ~70% recovered**
  - **Wh-traces: ~80% recovered**