Combining Labeled and Unlabeled Data in Statistical Natural Language Parsing

Dissertation Defense – 02/07/2002

Anoop Sarkar

Advisor: Prof. Aravind Joshi

Overview

- Tree Adjoining Grammars and Statistical Parsing

- Combining Labeled and Unlabeled Data in Statistical Parsing
  - Co-Training methods for statistical parsing
  - Learning unknown subcategorization frames
  - Learning verb alternations from minimally annotated corpora

- Conclusion

Overview

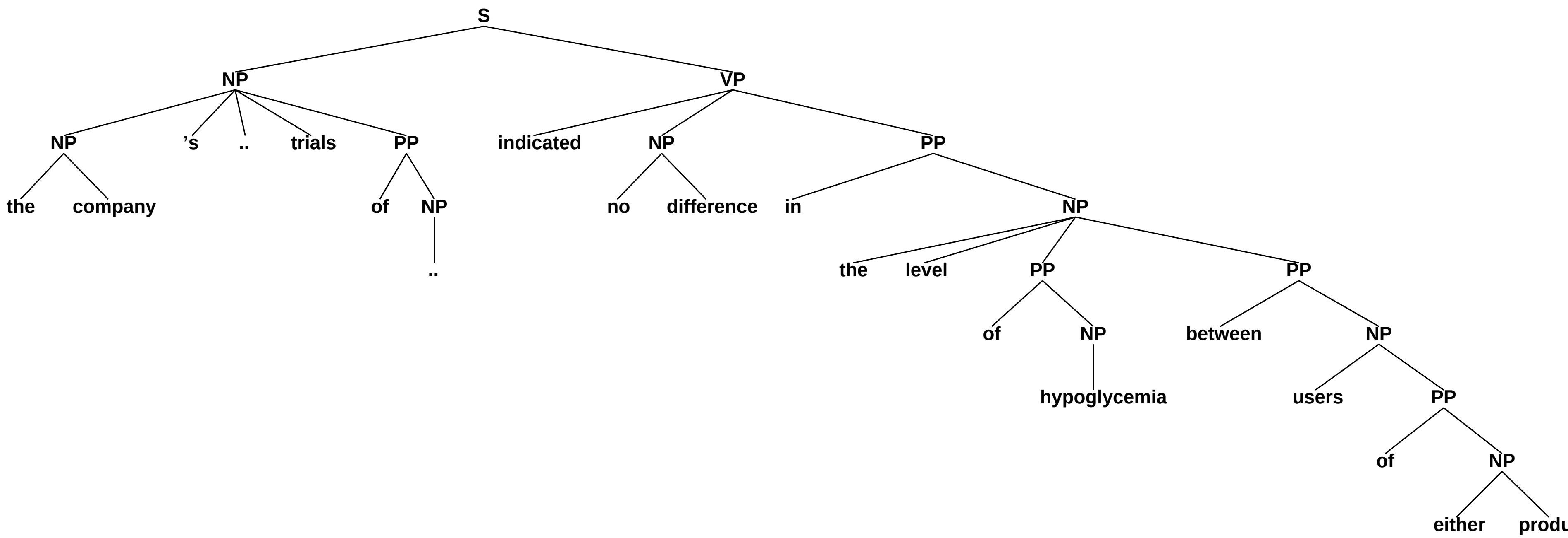
- $\boxed{\text{Tree Adjoining Grammars and Statistical Parsing}}$

- Combining Labeled and Unlabeled Data in Statistical Parsing
  - Co-Training methods for statistical parsing
  - Learning unknown subcategorization frames
  - Learning verb alternations from minimally annotated corpora

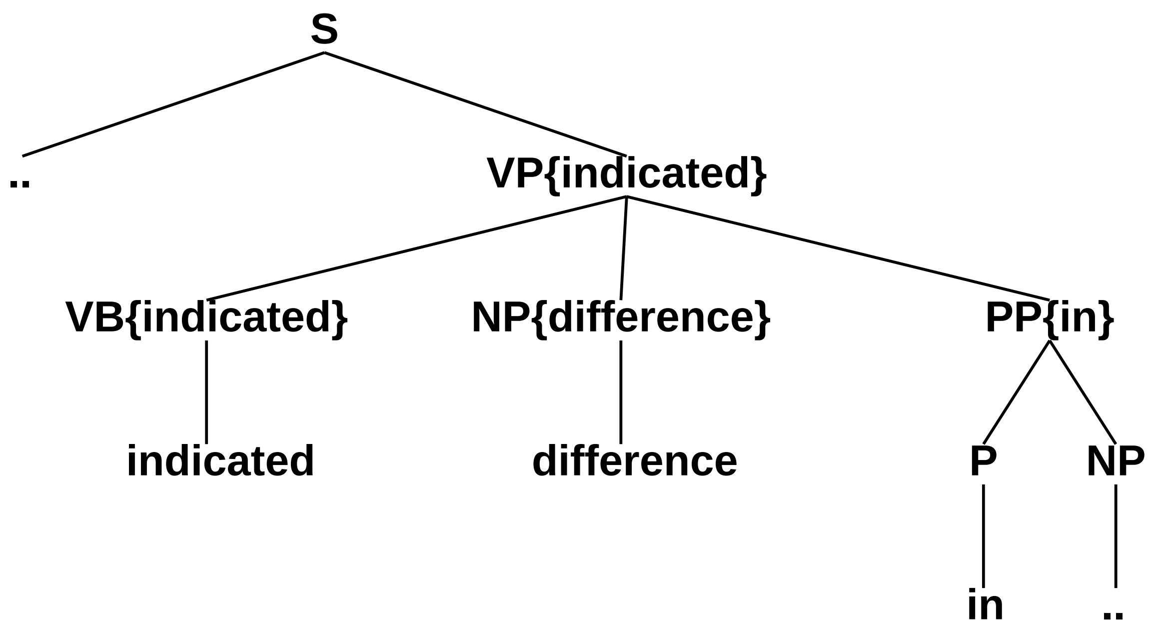
- Conclusion

Statistical Parsing:

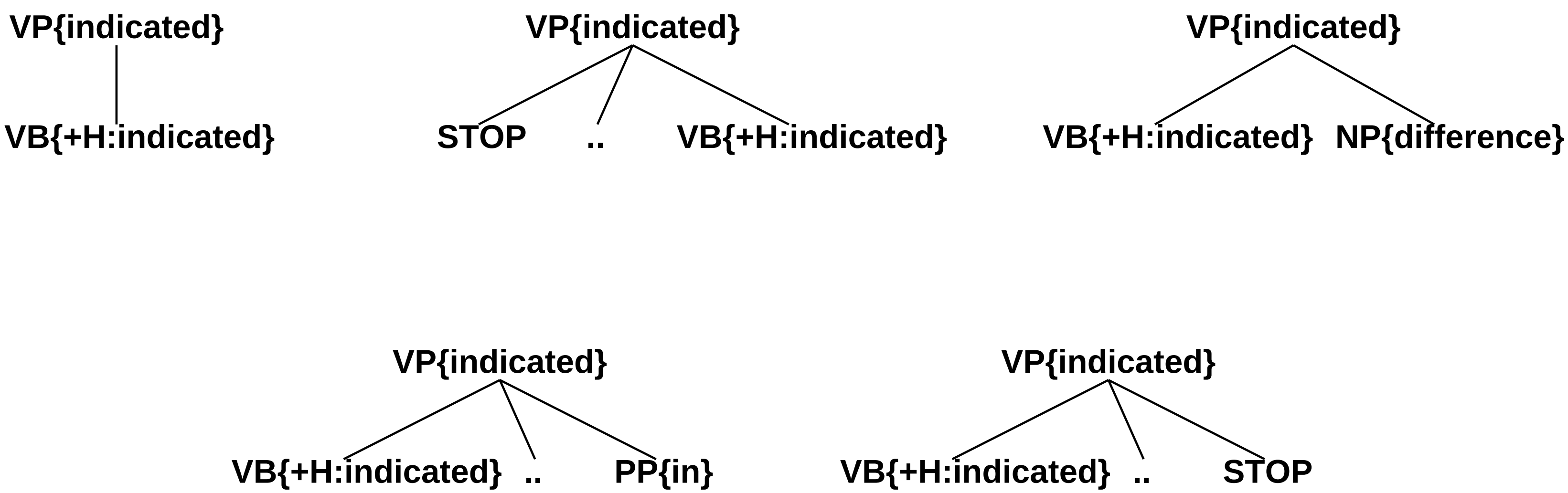
the company 's clinical trials of both its animal and human-based insulins indicated no difference in the level of hypoglycemia between users of either product



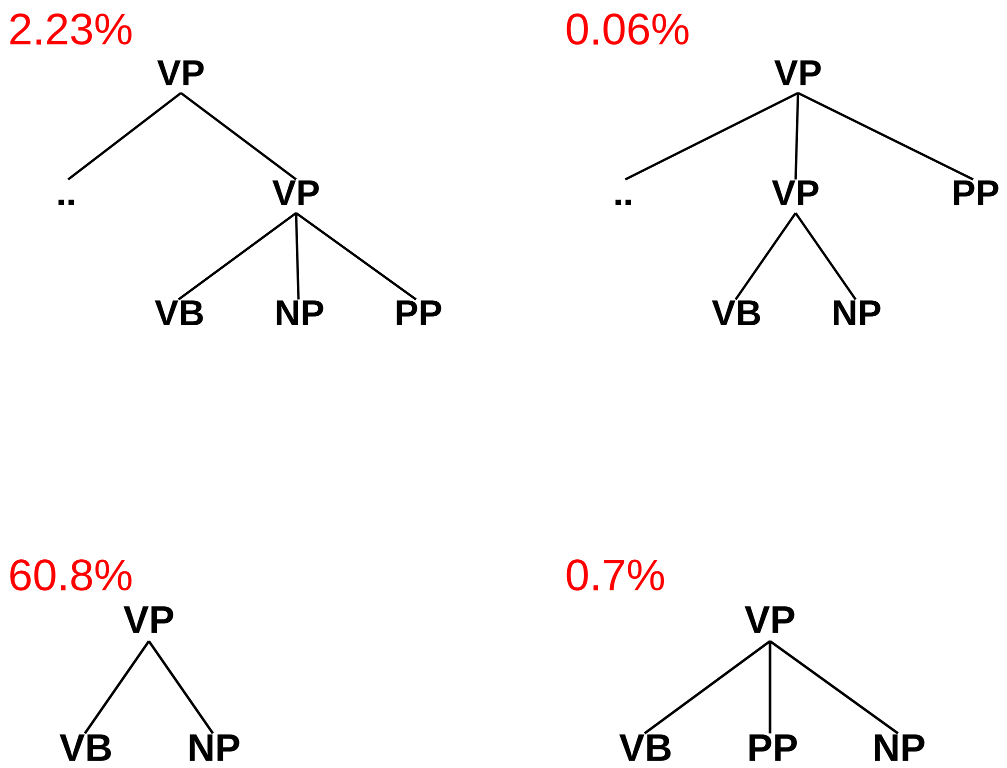
Lexicalized CFG



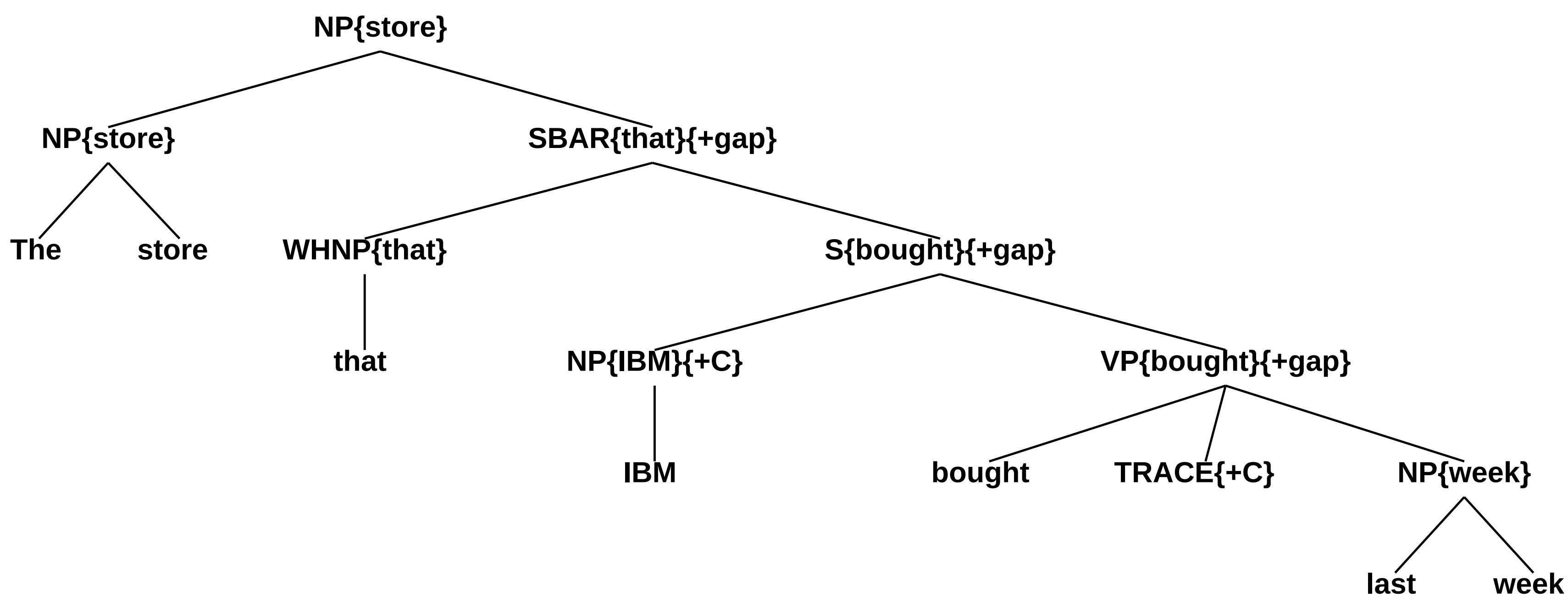
Bilexical CFG: $VP\{indicate\} \rightarrow VB\{+H{:}indicate\}\ NP\{difference\}\ PP\{in\}$



Independence Assumptions



Bilexical CFG with probabilistic ‘features’ (Collins 1999)



NP	$\rightarrow$	[	NP{+H}	SBAR{+gap}		]
SBAR{+gap}	$\rightarrow$	[	WHNP	S{+H}{+C}{+gap}		]
S{+gap}	$\rightarrow$	[	NP{+C}	SBAR{+H}{+gap}		]
VP{+gap}	$\rightarrow$	[	VB{+H}	TRACE{+C}	NP	]

Tree Adjoining Grammars

- Locality and independence assumptions are captured elegantly.

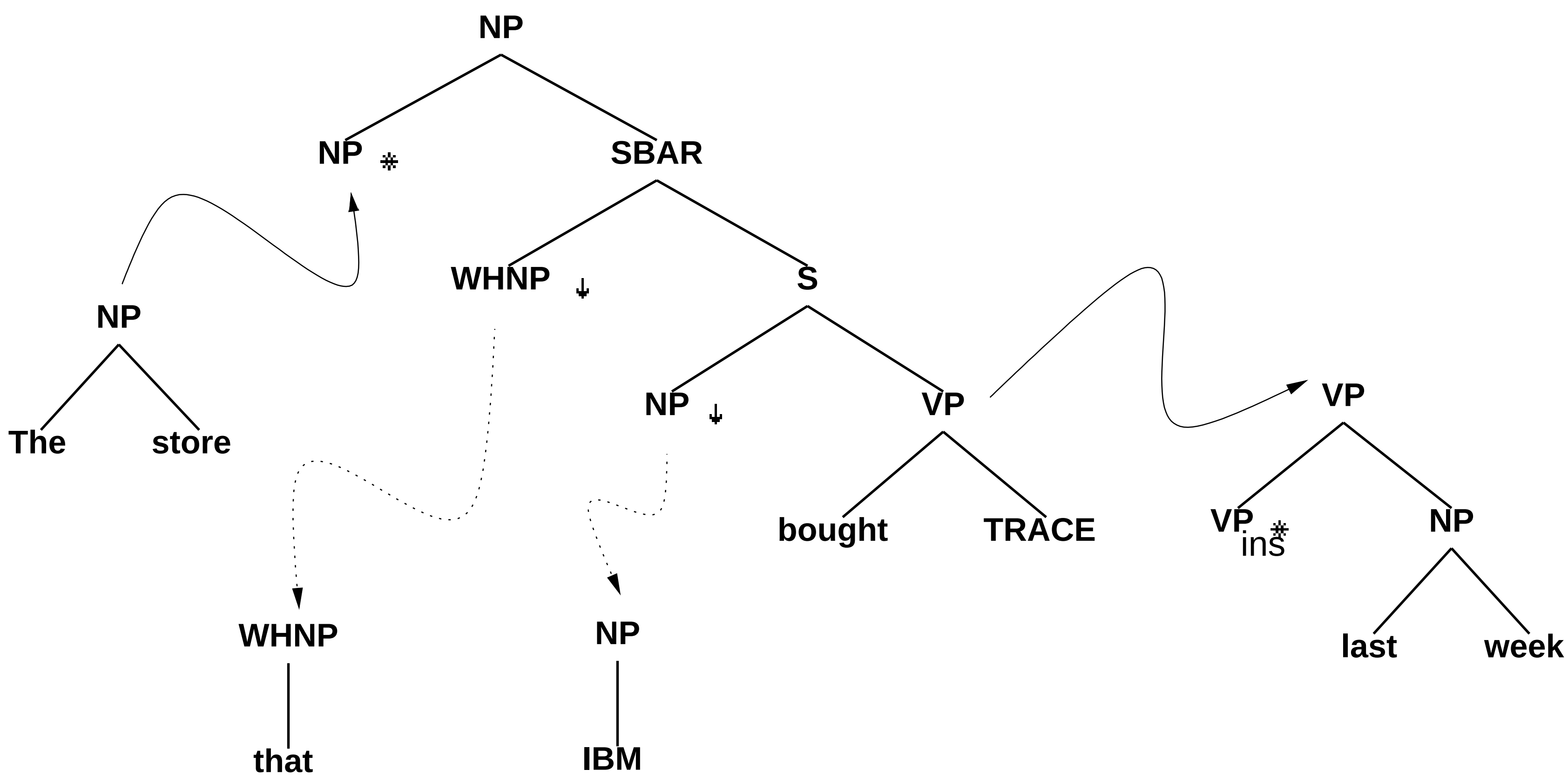
- Simple and well-defined probability model.

- Parsing can be treated in two steps:

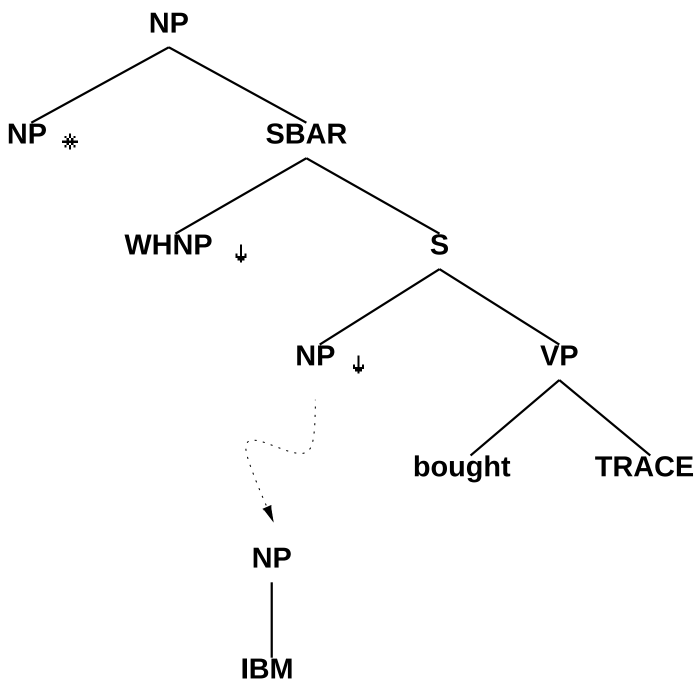
  1. Classification: structured labels (elementary trees) are assigned to each word in the sentence.

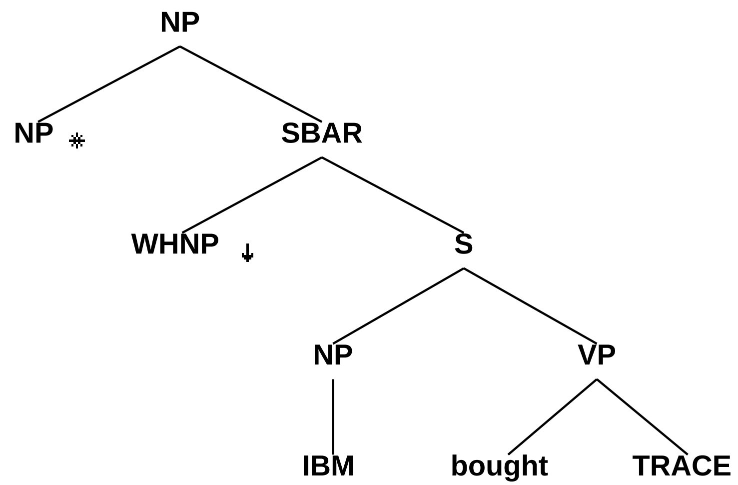
  2. Attachment: the elementary trees are connected to each other to form the parse.

Tree Adjoining Grammars: Different Modeling of Bilexical Dependencies



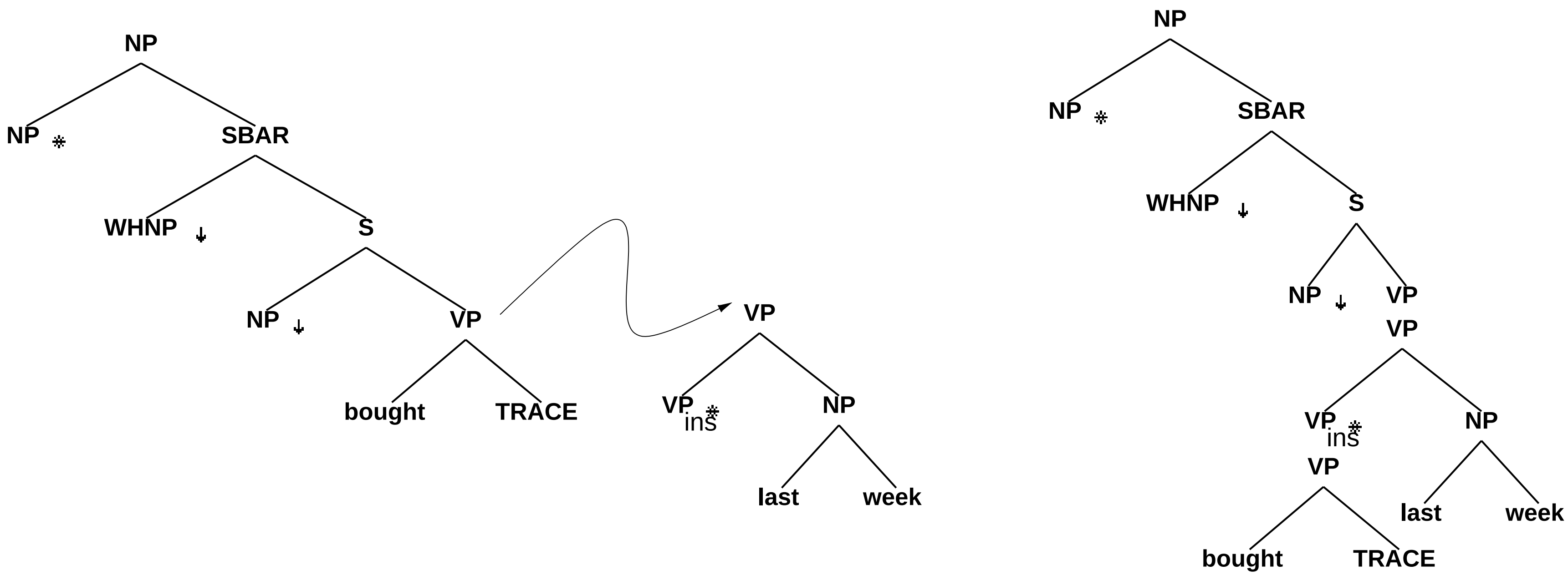
Probabilistic TAGs: Substitution





$$\sum_{\alpha} \mathcal{P}(t, \eta \rightarrow \alpha) = 1$$

Probabilistic TAGs: Adjunction



$$\mathcal{P}(t, \eta \rightarrow NA) + \sum_{\beta} \mathcal{P}(t, \eta \rightarrow \beta) = 1$$

Tree Adjoining Grammars

- Start of a derivation: $\sum_\alpha P_i(\alpha) = 1$

- Probability of a derivation:

$$
\begin{aligned}
& Pr(\mathcal{D}, w_0 \ldots w_n) = \\
& \quad P_i(\alpha, w_i) \times \prod_p P_s(\tau, \eta, w \rightarrow \alpha, w') \times \\
& \quad \prod_q P_a(\tau, \eta, w \rightarrow \beta, w') \times \prod_r P_a(\tau, \eta, w \rightarrow \mathsf{NA})
\end{aligned}
$$

Tree Adjoining Grammars

- Simpler model for parsing.
  Performance(Chiang 2000): 86.9% LR 86.6% LP ($\leq$ 40 words)

- Parsing can be treated in two steps:

  1. Classification: structured labels (elementary trees) are assigned to each word in the sentence.

  2. Attachment: Apply substitution or adjunction to combine the elementary trees to form the parse.

- Produces more than the phrase structure of each sentence.
  A more embellished parse in which phenomena such as predicate-argument structure, subcategorization and movement are given a probabilistic treatment.

Parsing as Classification and Attachment

- Assigning structured labels to each word results in an ‘*almost parse*’
  (Srinivas 1997)
  - A probabilistic treatment of classification: *SuperTagging*
  - A heuristic treatment of attachment:
    Lightweight Dependency Analyzer
- This work: a probabilistic treatment of both classification and attachment
- Extension to a more unsupervised approach
  (combining labeled and unlabeled data)

Theory of Probabilistic TAGs

PCFGs: (Booth and Thompson 1973); (Jelinek and Lafferty 1991)

- A probabilistic grammar is well-defined or consistent if:

$$\sum_{n=1}^{\infty} \sum_{a_1 a_2 \ldots a_n \in \mathcal{V}} \mathcal{P}(s \rightarrow a_1 a_2 \ldots a_n) = 1$$

- What is the single most likely parse (or derivation) for input string $a_1, \ldots, a_n$?

- What is the probability of $a_1, \ldots, a_i$, where $a_1, \ldots, a_i$ is a prefix of some string generated by the grammar? $\sum_{w \in \Sigma^*} P(a_1, \ldots, a_i w)$

- How should the parameters (e.g., rule probabilities) be chosen?

Overview

- Tree Adjoining Grammars and Statistical Parsing
- Combining Labeled and Unlabeled Data in Statistical Parsing
  - Co-Training methods for statistical parsing
  - Learning unknown subcategorization frames
  - Learning verb alternations from minimally annotated corpora
- Conclusion

Training a Statistical Parser

- How should the parameters (e.g., rule probabilities) be chosen?

- Alternatives:
  - EM algorithm: Inside-Outside Algorithm with labeled data
    (Schabes 1992; Hwa 1998)
  - Supervised training from a Treebank (Chiang 2000)
  - Parsing as Classification.
    Explore new machine learning techniques.

Statistical Parsing: Supervised vs. Unsupervised methods

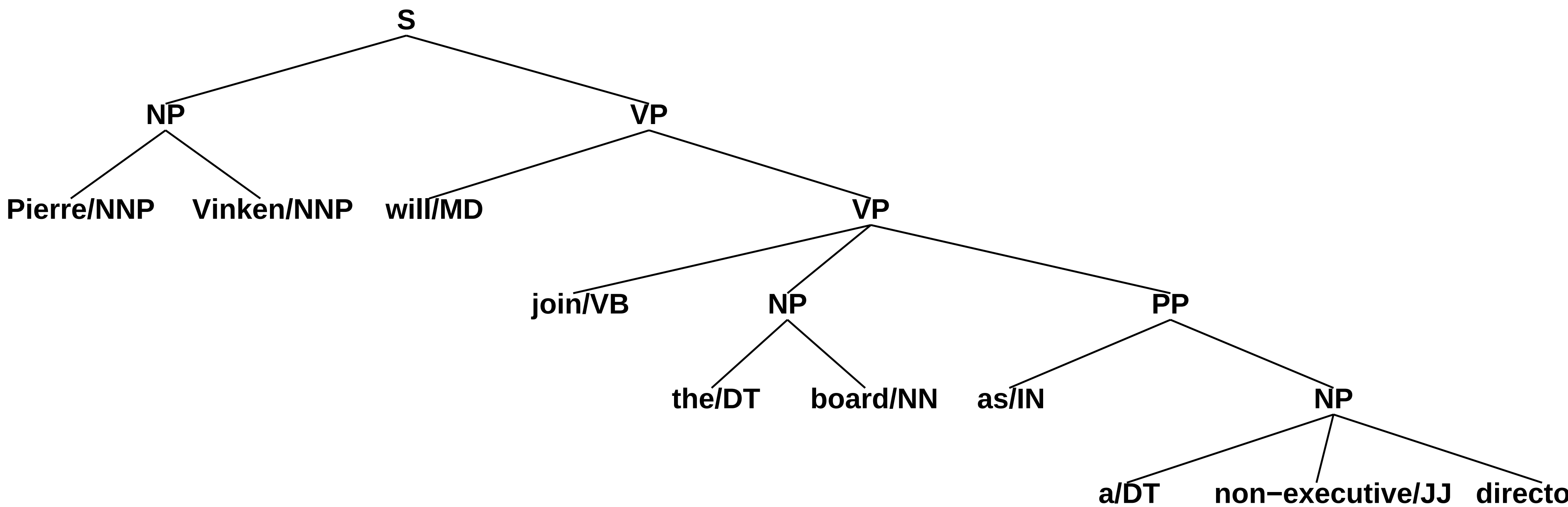
- Purely unsupervised approaches to parsing cannot handle structurally rich parses found in the Penn Treebank.
  (Lafferty et al 1992; Della Pietra et al 1994; de Marcken 1995)

- A feasible technique: Combining Labeled and Unlabeled Data
  - Active Learning: Bet on which examples are the hardest.
    (and annotate them) (Hwa 2000)
  - Co-Training: Bet on which examples can be handled with high confidence. (use as labeled data)

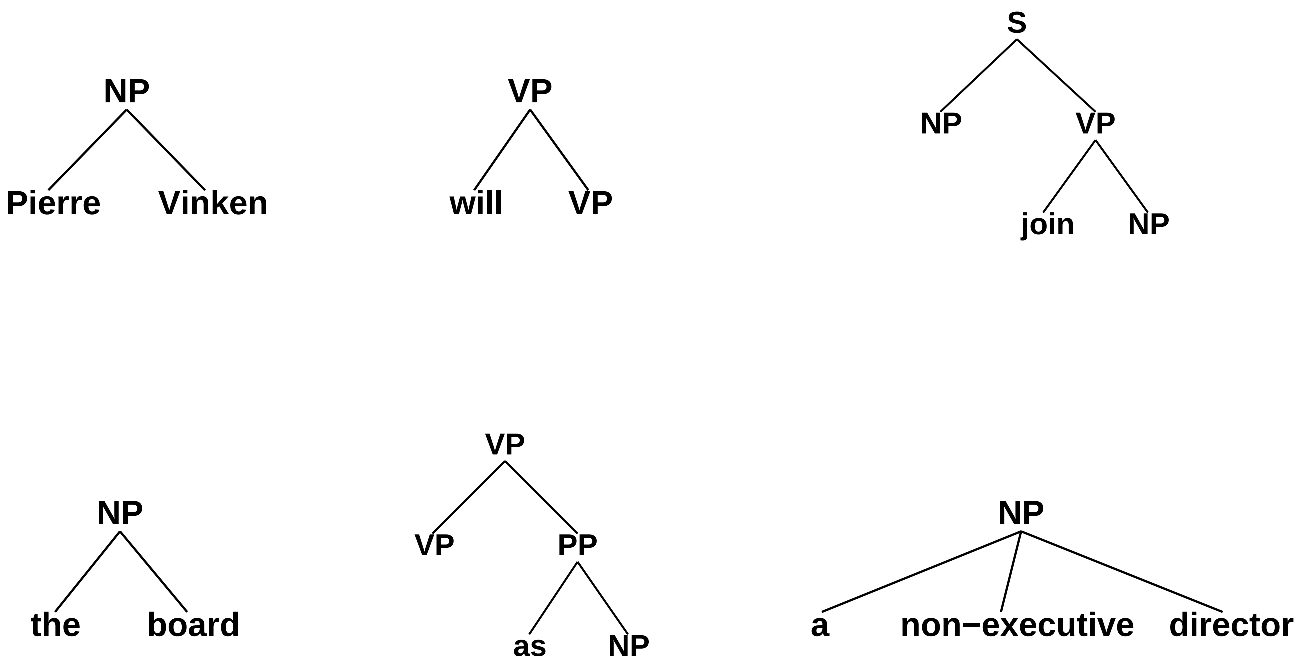
Co-Training (Blum and Mitchell 1998; Yarowsky 1995)

- Pick two “views” of a classification problem.
- Build separate models for each of these “views” and train each model on a small set of labeled data.
- Sample an unlabeled data set and to find examples that each model independently labels with high confidence. (Nigam and Ghani 2000)
- Pick confidently labeled examples.
  (Collins and Singer 1999; Goldman and Zhou 2000); Active Learning
- Each model labels examples for the other in each iteration.

Pierre Vinken will join the board as a non-executive director

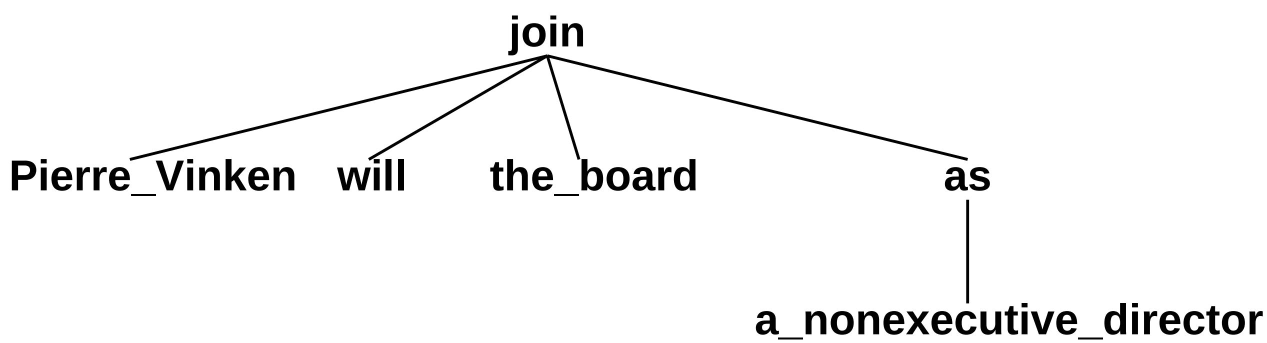


Parsing as Tree Classification and Attachment: (Srinivas 1997; Xia 2000)



Model H1: $\mathcal{P}(T_i \mid T_{i-2}T_{i-1}) \times \mathcal{P}(w_i \mid T_i)$

Parsing as Tree Classification and Attachment



Model H2: $\mathcal{P}(w, T \mid \text{TOP}) \times \Pi_i \mathcal{P}(w_i, T_i \mid \eta, w, T)$

The Co-Training Algorithm

1. Input: *labeled* and *unlabeled*
2. Update cache
   - Randomly select sentences from *unlabeled* and refill *cache*
   - If *cache* is empty; exit
3. Train models H1 and H2 using *labeled*
4. Apply H1 and H2 to cache.
5. Pick most probable n from H1 (stapled together) and add to *labeled*.
6. Pick most probable n from H2 and add to *labeled*
7. $n = n + k$; Go to Step 2

Results

- *labeled* was set to Sections 02-06 of the Penn Treebank WSJ (9625 sentences)

- *unlabeled* was 30137 sentences (Section 07-21 of the Treebank stripped of all annotations).

- A tree dictionary of all lexicalized trees from *labeled* and *unlabeled*.
  Similar to the approach of (Brill 1997)
  Novel trees were treated as unknown tree tokens

- The *cache* size was 3000 sentences.

Results

- Test set: Section 23
- Baseline Model was trained only on the *labeled* set:

  Labeled Bracketing Precision = 72.23% Recall = 69.12%
- After 12 iterations of Co-Training:

  Labeled Bracketing Precision = 80.02% Recall = 79.64%
- Evaluation of an unsupervised approach is directly comparable to other supervised parsers (unlike previous work).

Co-Training and EM

	max likelihood over full unlabeled set	iterative selection from unlabeled set
$Q_0 \mid\mid Q_\infty$	EM†	self-training
conditionally independent features	co-EM*	Co-Training

* (Nigam and Ghani, 2000)

† Discriminative Objective f; $Q_0 \mid\mid Q_{dis}$ (Mitchell, to appear)

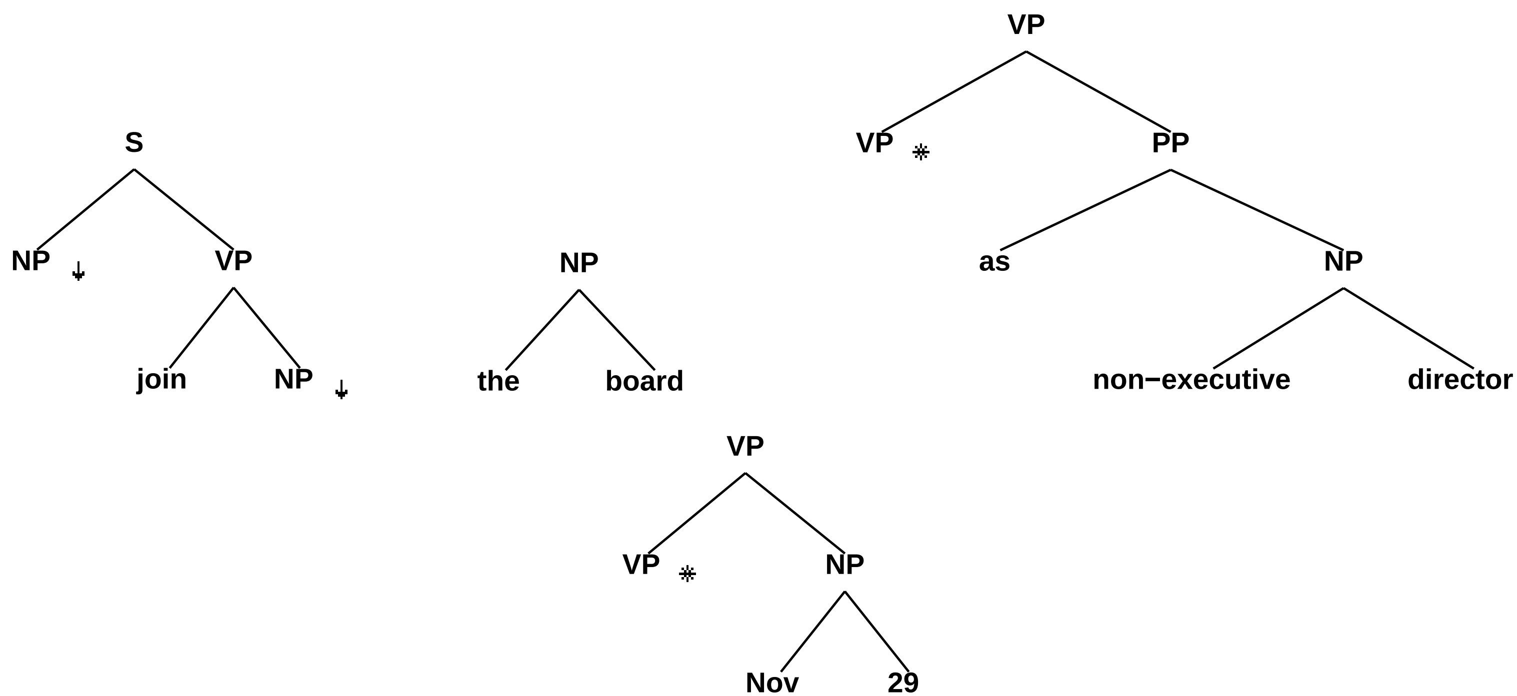
Experiments with larger labeled sets

- Still needs human supervision to create the tree dictionary.
  For small datasets, this is unavoidable.

- Another application: use a large labeled dataset
  But improve performance using a much larger unlabeled dataset.

- Expt: 1M words *labeled* and 23M words *unlabeled*.
  Tree dictionary is completely defined by the labeled set.

Co-training with two parsers



- Two different probability models for adjunction: single vs. multiple adjunction
- Non-overlapping lexicalized features: ⟨*join, Nov_29*⟩ vs. ⟨*as, Nov_29*⟩.

Co-training with two parsers

- Trained two parsers using these two models on sections 02-21 of the Penn Treebank.
- We then performed co-training using a larger set of WSJ unlabeled text (23M words).
- Even after 12 iterations of co-training, performance did not improve significantly over the baseline of LR 85.2% and LP 86%.

Possible Reasons why Co-training did not improve performance significantly

- Reason 1: 1M words of data is enough for current models;
  ⇒ unlikely: because bigrams/unigrams in parsing models lead to sparse data problems. cf. (Gildea 2001)

- Reason 2: The tag dictionary was incomplete;
  ⇒ unlikely: because of lack of parsing failures and performance remained close to baseline

- Reason 3: Substantial overlap between the features used in each of the probability models;
  ⇒ **likely: only 22% of the lexicalized features were different**

Future Work with Co-training

- Co-training multiple parsers (JHU workshop 2002)
- Discriminative methods with unlabeled data:
  max likelihood (EM) vs. sampling estimation of error reduction
- Combining with voting methods: adding constituents

Overview

- Tree Adjoining Grammars and Statistical Parsing
- Combining Labeled and Unlabeled Data in Statistical Parsing
  - Co-Training methods for statistical parsing
  - Learning unknown subcategorization frames
  - Learning verb alternations from minimally annotated corpora
- Conclusion

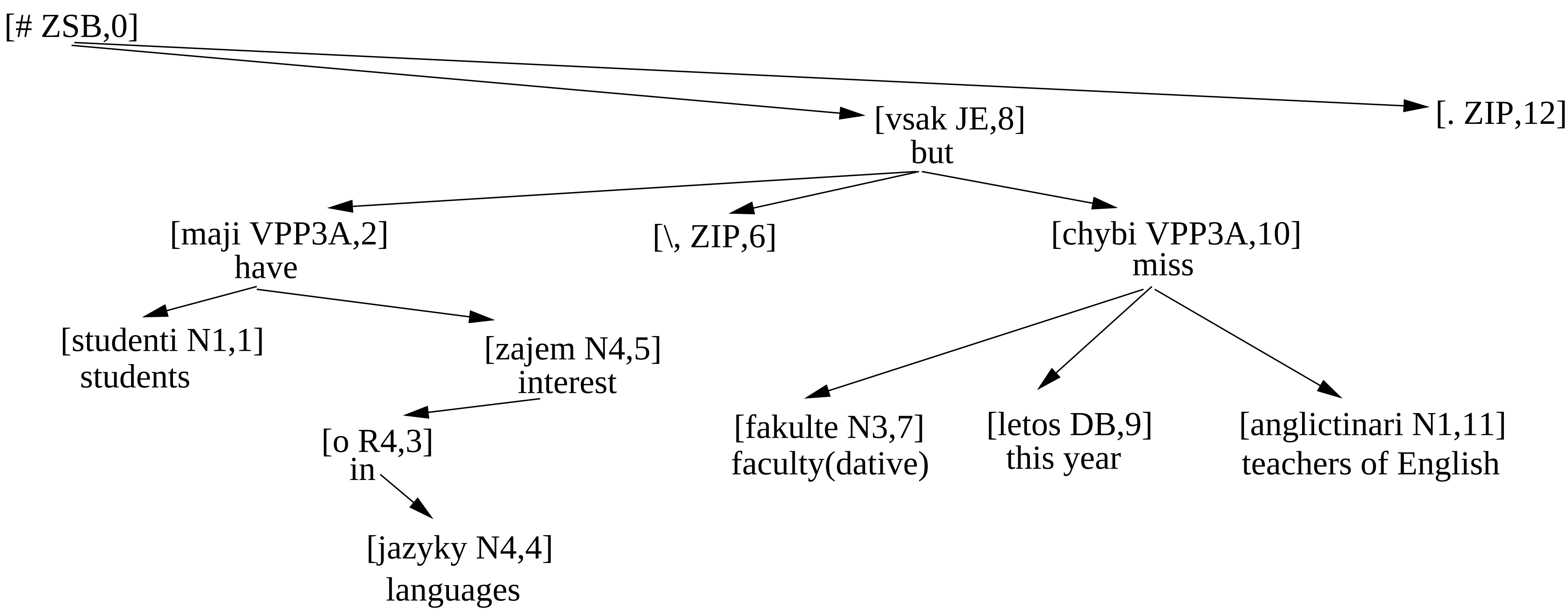
The Task

- Discover valid subcategorization frames (SFs) for each verb
- Distinguish arguments from adjuncts
- Learning from data *not* annotated with SF information

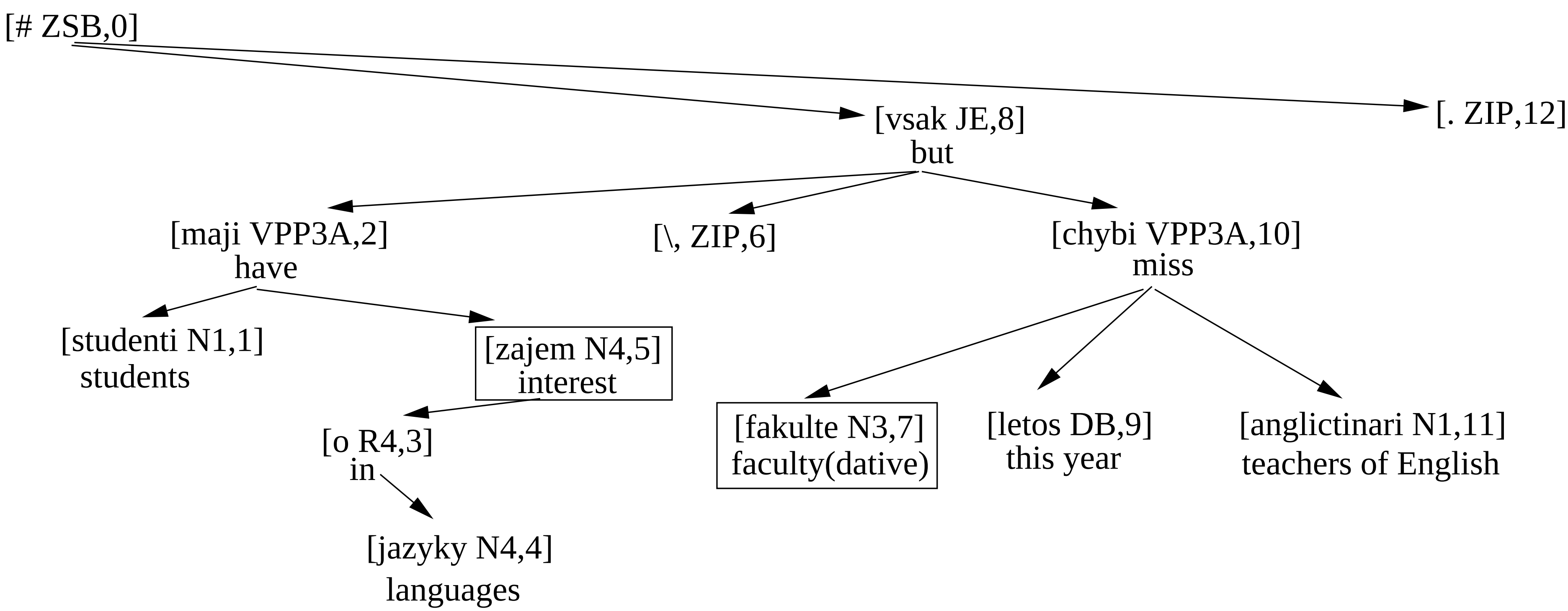
Comparison to previous work

Previous Work	**Current Work**
Predefined set of SFs	SFs are learned from data
Learning from parsed or chunked data	Adds SF information to an existing treebank
Difficult to add info to existing treebank parser	Existing treebank parser can easily use SF info
Most work done on English	Czech

Prague Dependency Treebank



Annotation Provided by Algorithm



Argument Types: lexicalized SFs

- Noun phrases: N4, N3, N2, N7, N1
- Prepositional phrases: R2(bez), R3(k), R4(na), R6(na), R7(s), . . .
- Reflexive pronouns *se*, *si*: PR4, PR3
- Clauses: S, JS(že), JS(zda)
- Infinitives (VINF), passive participles (VPAS), adverbs (DB)

Methods Used

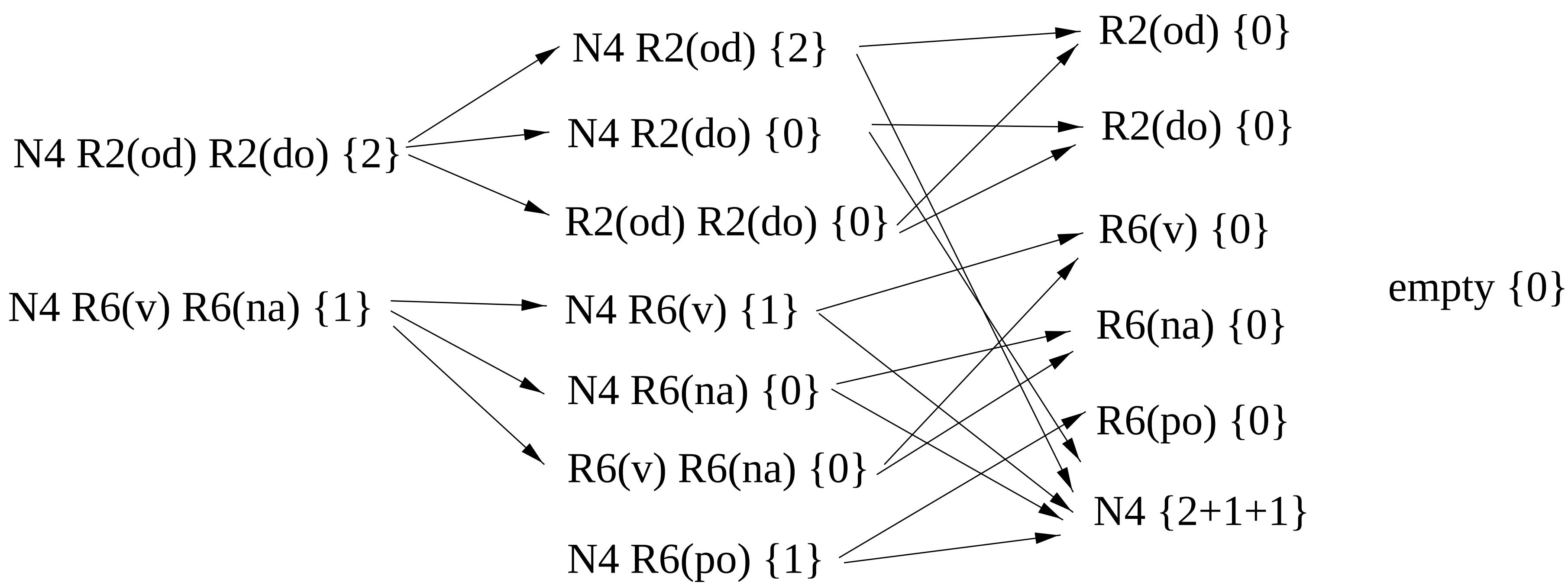
- Hypothesis Testing using:
  - Likelihood Ratio test
  - T-score test
  - Binomial models of miscue probabilities

- Hypothesis: $\underbrace{p(f \mid v)}_{p_1} = \underbrace{p(f \mid !v)}_{p_2} = \underbrace{p(f)}_{p}$

Subsets of observed frames

- Iterative algorithm:
  - First use counts for the observed frame f in hypothesis testing
  - If f is rejected as true SF, produce all subsets of f
  - Select one subset of f as successor observed frame s which is updated with f's counts
  - Repeat for each s rejected by hypothesis testing

Subsets of observed frames



Successor Selection

1. Choose the successor frame that results in the strongest preference (lowest entropy across the corpus; exponential in num of frames)

2. Pick the successor frame with highest cumulative frequency at each step (greedy)

3. Random selection

$\rightarrow$ *Random selection works the best*

Baseline methods

- Baseline method 1: consider each dependent of a verb an adjunct.
- Baseline method 2:
  - Use the longest known observed frame matching the test candidate.
  - If no matching OF, find longest partial match.
  - Exploit functional and morphological tags while matching.
- No statistical filtering is applied in either baseline method.

Results

- 19,126 sents (300K words) training data
- 33,641 verb tokens; 2,993 verb types; 28,765 observed frames
- 13,665 frames after omitting clear adjuncts
- 914 verbs seen $>$ 5 times
- 137 frame classes learned
- Test data: 495 sentences annotated by hand

Results

	Baseline 1	Baseline 2
Precision	55%	78%
Recall:	55%	73%
$F_{\beta=1}$	55%	75%
% unknown	0%	6%

	Lik. Ratio	T-scores	Miscue Rate
Precision	82%	82%	88%
Recall:	77%	77%	74%
$F_{\beta=1}$	79%	79%	80%
% unknown	6%	6%	16%

Comparison with Previous Work

Previous work	Data	#SFs	#verbs tested	Method	Miscue rate	Corpus
Ushioda93	POS + FS rules	6	33	heuristics	NA	WSJ (300K)
Brent93	raw + FS rules	6	193	Hypothesis testing	iterative estimation	Brown (1.1M)
Manning93	POS + FS rules	19	3104	Hypothesis testing	hand	NYT (4.1M)
Brent94	raw + heuristics	12	126	Hypothesis testing	non-iter estimation	CHILDES (32K)
Ersan96	Full parsing	16	30	Hypothesis testing	hand	WSJ (36M)
Briscoe97	Full parsing	160	14	Hypothesis testing	Dictionary estimation	various (70K)
Carroll98	Unlabeled	9+	3	Inside-outside	NA	BNC (5-30M)
Current	Fully Parsed	Learned 137	914	Subsets+ Hyp. testing	Estimate	PDT (300K)

Overview

- Tree Adjoining Grammars and Statistical Parsing
- Combining Labeled and Unlabeled Data in Statistical Parsing
  - Co-Training methods for statistical parsing
  - Learning unknown subcategorization frames
  - Learning verb alternations from minimally annotated corpora
- Conclusion

Classification of Verb Alternations: Application of SF Learning

Unergative

INTRAN: The horse raced past the barn. (NP_{agent} raced)

TRAN: The jockey raced the horse past the barn. (NP_{causer} raced NP_{agent})

Unaccusative

INTRAN: The butter melted in the pan. (NP_{theme} melted)

TRAN: The cook melted the butter in the pan. (NP_{causer} melted NP_{theme})

Object-Drop

INTRAN: The boy washed. (NP_{agent} washed)

TRAN: The boy washed the hall. (NP_{agent} washed NP_{theme})

(Stevenson and Merlo 1997)

The Hypothesis (Merlo and Stevenson 2001)

- All verbs in each class can occur with the same syntactic context as other verbs
- Statistical distributions of syntactic context can be distinguished for each verb
- Identify probabilistic features that pick out verb co-occurences with particular syntactic contexts and use for classification
- This work: application of SF learning to this kind of classifier to see if noisy data with less annotation can be used

Corpus tagged by Adwait Ratnaparkhi's tagger and then chunked using Steve Abney's chunker:

Pierre	NNP	nx	2
Vinken	NNP		
,	,		
61	CD	ax	3
years	NNS		
old	JJ		
,	,		
will	MD	vx	2
join	VB		
the	DT	nx	2
board	NN		
as	IN		
a	DT	nx	3
nonexecutive	JJ		
director	NN		
Nov.	NNP		
29	CD		
.	.		

Features used (cf. Merlo and Stevenson 2001)

1. simple past (VBD), and past participle(VBN)
2. active (ACT) and passive (PASS)
3. causative (CAUS)
4. animacy (ANIM)

- POS features: part of speech of subject and object head noun
- SF features: transitive (TRAN) and intransitive (INTRAN)

Results

- Data: 23M words of WSJ text chunked
- 76 verbs picked to balance frequency (classes from Levin)
- Baseline: pick argument structure at random, ER = 65.5%
- (Merlo and Stevenson 2001) measure expert-based upper bound, ER = 13.5%
- (Merlo and Stevenson 2001) obtain ER = 30.2% with 65M words of automatically parsed WSJ text
- Current work: C5.0 classifier (using SF info), ER = 33.4% with 23M words of chunked text (SF info obtained by learning)

Overview

- Tree Adjoining Grammars and Statistical Parsing
- Combining Labeled and Unlabeled Data in Statistical Parsing
  - Co-Training methods for statistical parsing
  - Learning unknown subcategorization frames
  - Learning verb alternations from minimally annotated corpora
- Conclusion

Contributions of the Dissertation

- Theoretical Work (not presented in this talk)
  - Consistency of Probabilistic TAGs
  - Prefix Probabilities from Probabilistic TAGs
  - Head-corner parsing algorithm for TAGs (implementation used in XTAG)
- Corpus-Based Work (combining labeled and unlabeled data)
  - Co-Training methods for statistical parsing.
  - Learning unknown subcategorization frames.
  - Learning verb alternations from minimally annotated corpora.

Future Directions

- Combining multiple parsers with the use of unlabeled data
  - Co-training multiple parsers (JHU workshop 2002)
  - Discriminative methods with unlabeled data:
    max likelihood (EM) vs. sampling estimation of error reduction
  - Combining with voting methods: adding constituents
- Statistical Parsing
  - Smoothing a PCFG and finding heads: SF subset algorithm. cf. (Eisner 2001)
  - SF learning and verb alternation features for parsing
  - Multilingual statistical parsing: English, Korean, Hindi, Chinese, Czech, Arabic