CMPT-882: Statistical Learning of Natural Language

Lecture #9

Anoop Sarkar
anoop@cs.sfu.ca
http://www.sfu.ca/~anoop

- *Automatic Extraction of Subcategorization from Corpora*. Ted Briscoe and John Carroll. 1997.

- *Accurate Methods for the Statistics of Surprise and Coincidence*. Ted Dunning. Computational Linguistics. 1993.

- *Surface Cues and Robust Inference as a Basis for the Early Acquisition of Subcategorization Frames*. Michael Brent. 1993.

- *Automatic Extraction of Subcategorization Frames for Czech*. Anoop Sarkar and Daniel Zeman. 2000.

Subcategorization Frames

- Certain words like *verbs* have required semantic arguments

  1. *Ted watched baseball* $\rightarrow$ WATCH(E, TED, BASEBALL) & PAST(E)

- Different words have different number of required arguments

  1. *Ted watched baseball*

  2. $*$ *Ted looked baseball*

Subcategorization Frames

- Another example: *pretend* vs. *play*

  1. *Juan is pretending* { *to be grown up*}
  2. ∗ *Juan is playing* { *to be grown up* }
  3. *Juan is pretending* { *that he is grown up* }
  4. ∗ *Juan is playing* { *that he is grown up* }

- Chomsky (1965) referred to these properties of words as their *subcategorization properties* (i.e. properties that distinguish a word from others in its syntactic category)

Subcategorization Frames

- Just as a word can have multiple categories, it can also have multiple subcategorization frames
- We used part of speech tagging techniques to find the right syntactic category for a word
- Unfortunately, there is no labeled data that we can use to train a *subcat* tagger
- So we are forced to bootstrap this information automatically

Subcategorization Frames

- Interesting question for language acquisition: does meaning let us learn syntax, or the other way around?
- Let us assume that the set of subcat frames is fixed in advance
- We want to learn the mapping between words and some subset of this set of frames
- Similar to finding the *tag dictionary* for part of speech tagging

Subcategorization Frames

- Perhaps we can just count cues from the input and assign observed frames to a word

  1. *Juan pretends to fish*

  2. *Juan drove to fish markets all over town*

- Can we find the exceptions automatically?

SF Description	Good Example		Bad Example
NP only	greet them	*	arrive them
tensed clause	hope he'll attend	*	tell he'll attend
infinitive	hope to attend	*	greet to attend
PP only	listen to me	*	put on it
NP & clause	tell him he's a fool	*	yell him he's a fool
NP & infinitive	want him to attend	*	hope him to attend
NP & NP	tell him the story	*	want him the story
NP & PP	put it on the table	*	listen him to me

Subcategorization Frames for verbs

- Why only verbs?

  1. *Juan liked to pretend he saw a unicorn* VERB

  2. $*$ *Juan liked to play he saw a unicorn* VERB

  3. *Juan liked the play he saw with a unicorn* NOUN

- Nouns can be modified by clauses, verbs are distinct in their usage.

Robust inference

- Data: collect for each verb, every frame observed with the verb

- We can use chunking techniques we have already seen to find text chunks around the verb

  1. $\{_N$ *Juan* $\}$ $\{_{VG}$ *is going to give* $\}$ $\{_N$ *Jorge* $\}$ $\{_N$ *a map* $\}$

  2. $\{_N$ *Juan* $\}$ $\{_{VG}$ *donated* $\}$ $\{_N$ *a map* $\}$ $\{_{PP}$ *to the library* $\}$

- Not all of these frames will be true subcat frames

Robust inference

- Let's call the event $f \mid v$ when a frame f is seen with a verb v and it **is** a true subcat frame

- And $f \mid !v$ is an event when a frame f is seen with v and **is not** a true subcat frame

- $f \mid !v$ is called a *miscue*

- Miscues occur because of chunking errors or because cues can include non-arguments,
  e.g. *John provided for his family* vs. *John ate with his hands*

Robust inference

- Let us assume that for a fixed frame f that all verbs v such that $f\ |!v$ have the same probability for a miscue
- Let $p(f\ |!v)$ be this probability which varies from frame to frame but does not vary from verb to verb
- Once we know the miscue rate, we can use hypothesis testing to distinguish miscues for a particular verb
- Hypothesis: $\underbrace{p(f \mid v)}_{p_1} = \underbrace{p(f \mid !v)}_{p_2} = \underbrace{p(f)}_{p}$

Robust inference

- For example, let's say that $p(f\ |!v) = 0.05$
- Let's say we see a verb v 300 times; of which we see v with a frame f 30 times
- Event $f\ |!v$ is not close to the miscue probability of 0.05
- Hence f is unlikely to be a miscue for v in this case
- for each verb such that $f \mid v$, the prob $p(f \mid v)$ is going to be greater than the miscue probability

Robust inference

- Two verbs seen with the same frame can have different probabilities of occurring with that frame, as long as this prob is greater than the miscue prob

- How do we decide based on the counts of frames observed with a verb, whether it falls above the miscue rate

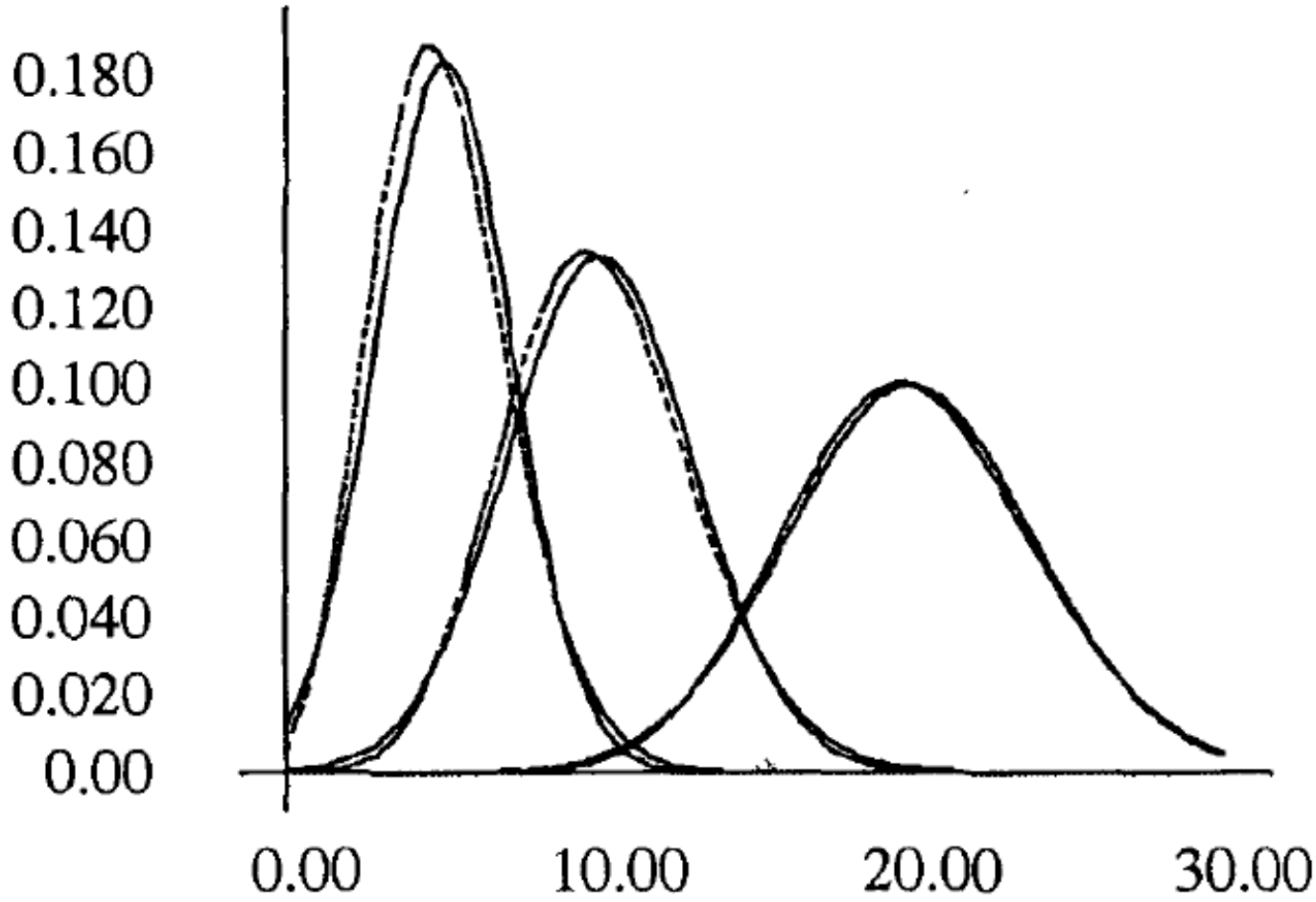
- If a coin prob p of flipping heads, and it's flipped n times, the prob that it comes up heads m times is given by the binomial distribution

$$Pr(m, n, p) = p^m (1-p)^{n-m} \binom{n}{m}$$

Binomial Distribution

$$Pr(m, n, p) = p^m(1-p)^{n-m}\binom{n}{m}$$

- mean is np; variance is $np(1-p)$


0.180
0.160
0.140
0.120
0.100
0.080
0.060
0.040
0.020
0.00
0.00
10.00
20.00
30.00

Why not use the normal distribution

	Using binomial	Using normal
$np = 0.001$	0.000099	0.34×10^{-219}
$np = 0.01$	0.0099	0.29×10^{-22}
$np = 0.1$	0.095	0.0022
$np = 1$	0.63	0.5

Collocation analysis using Binomial vs Normal

- Bigrams AB; Counts $k(AB)$; $k(!AB)$; $k(A!B)$; $k(!A!B)$

- We want to find sticky words or collocations; hypothesis is that is words occur independently then:

$$p(A \mid B) = p(A \mid !B) = p(A)$$

Binomial		Normal	
the	swiss	natel	c
can	be	write	offs
previous	year	wood	pulp
⋮		⋮	
a	positive	appenzell	abrupt

Robust inference

$$Pr(m, n, p) = p^m(1-p)^{n-m}\binom{n}{m}$$

$$Pr(m, n, p) = p^m(1-p)^{n-m}\frac{n!}{n!(n-m)!}$$

- The prob of heads coming up m times or *more* is:

$$Pr(m+, n, p) = \sum_{i=m}^{n} P(i, n, p)$$

- Take $Pr(m+, n, p)$ to be the prob that m or more occurrences of a verb v where f $|!v$ (i.e. frame f is not a subcat frame for v) will occur with frame f out of n occurrences in the data

Robust inference

- $Pr(m+, n, p)$: prob that given n occurrences of a verb, m or more occurrences of a frame will occur with a verb that does not take that frame

- To use this value, we set a threshold value

- A threshold of less than equal to 0.05 yields a 95% or better confidence level that a frame is indeed a subcat frame for a verb

Estimating the miscue probability

- We still need to find the miscue rate for the hypothesis testing method
- In Briscoe and Carroll (1997), the miscue rate is estimated from a dictionary
- Remember: this probability which varies from frame to frame but does not vary from verb to verb

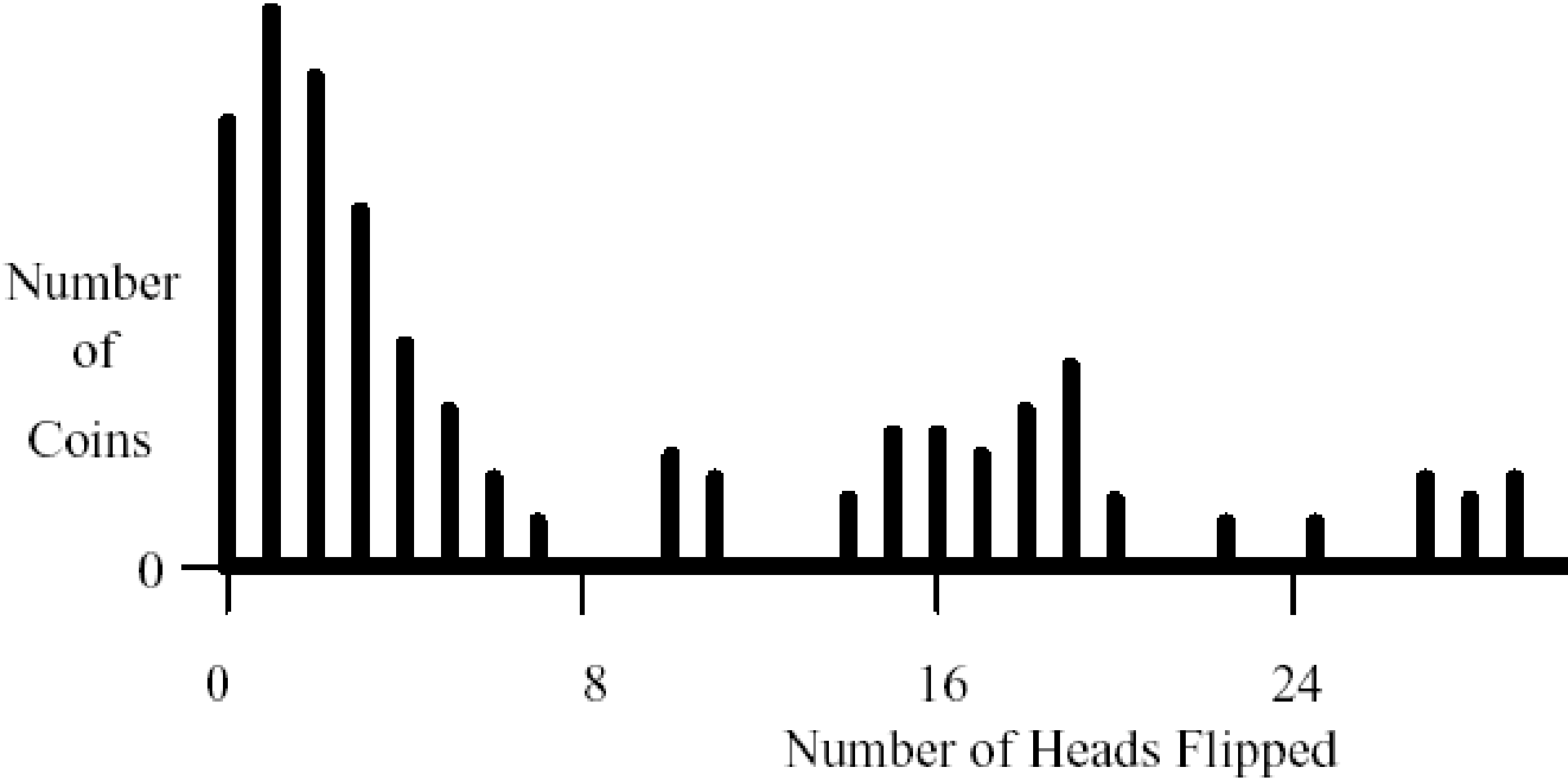
Estimating the miscue probability: Briscoe and Carroll (1997)

- Let V be the set of all the verbs in a corpus seen in a dictionary
- Let V_f be the set of all verbs associated with frame f in the dictionary
- Let $\text{freq}(f)$ be the frequency of frame f and let $N = \sum_f \text{freq}(f)$

$$p(f \mid !v) = (1 - \frac{\mid V_f \mid}{\mid V \mid}) \times \frac{\text{freq}(f)}{N}$$

Estimating the miscue probability: Brent (1993)

- Take a set of coins and flip it N times
- Now put each coin into a bin representing how many times it came up as heads
- bin i contains the number of coins that came up heads i times out of N
- we want to find coins biased against coming up heads but have a certain rate of coming up as heads: $p(h \mid !c)$


Number
of
Coins
0
0
8
16
24
Number of Heads Flipped

Estimating the miscue probability: Brent (1993)

- Coins whose prob of turning up heads is a miscue prob should be clustered at the start of the histogram (with the low heads values)

- So there should be a j_0, where $0 \leq j_0 \leq N$ where average rate of coins in bin j_0 or less is equal to the miscue rate

- The shape of the curve below j_0 should be a binomial distribution, separate from the coins that have different bias

Estimating the miscue probability: Brent (1993)

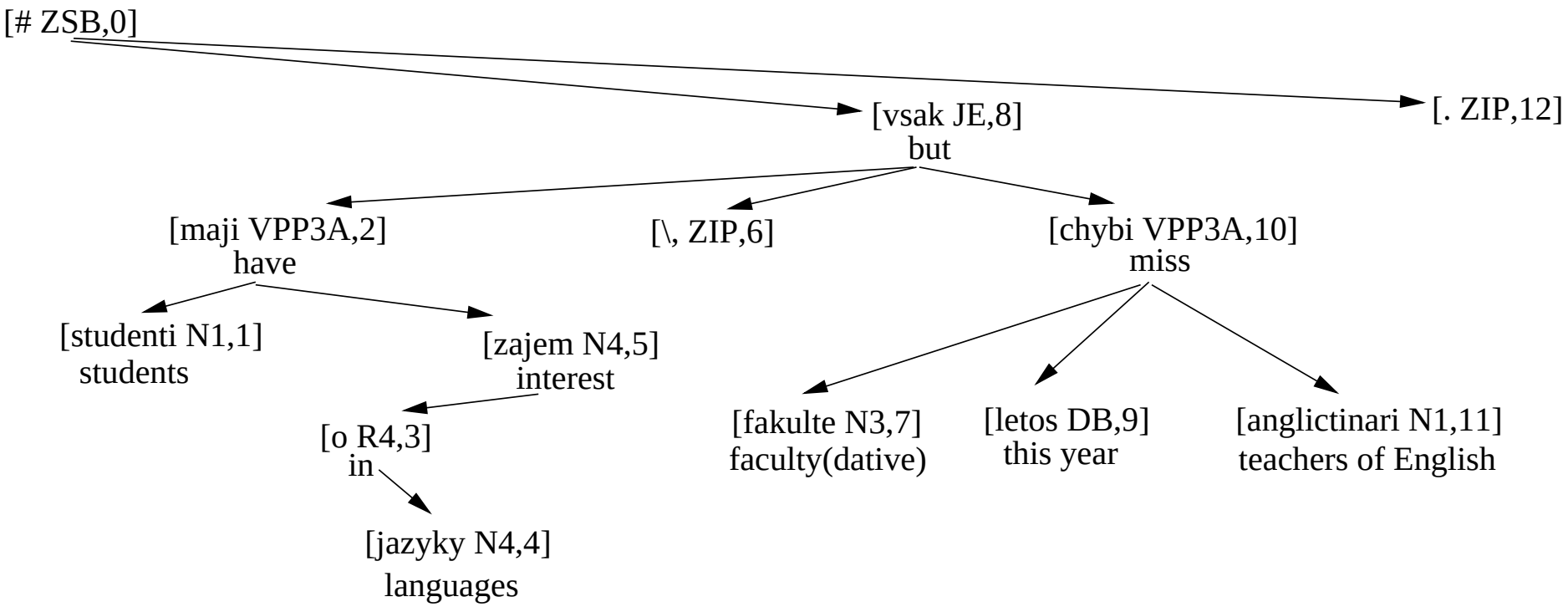
- Algorithm: slide a window and match a binomial to the points to find an appropriate j_0 value

- Average number of times a coin came up heads for those coins below j_0 gives us the miscue rate

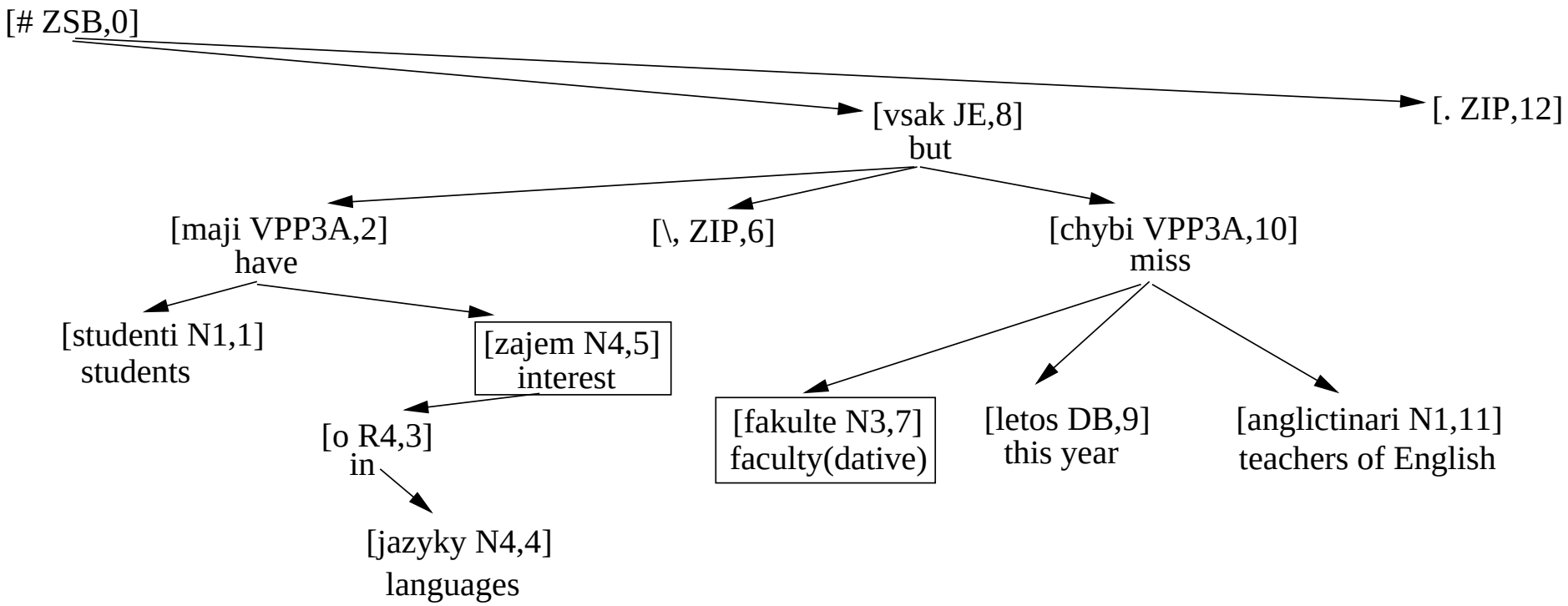
Comparison to previous work

Previous Work	**Sarkar and Zeman (2000)**
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

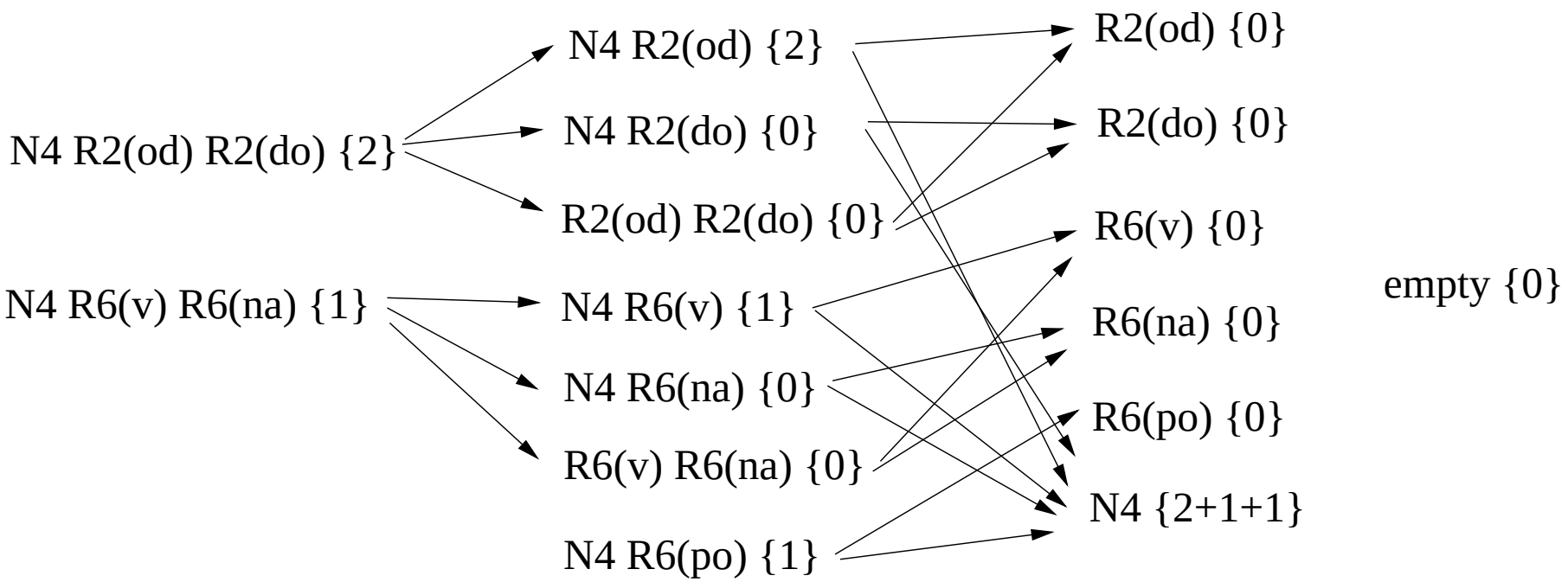
- Hypothesis Testing using:
  - Likelihood Ratio test
  - T-score test
  - Binomial models of miscue probabilities

- 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

→ *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%

Summary of Work in Subcat Learning

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)

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
- 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%

Results

- (Merlo and Stevenson 2001) obtain ER = 30.2%
  with 65M words of automatically parsed WSJ text

- C5.0 classifier (using SF info), ER = 33.4%
  with 23M words of chunked text (SF info obtained by learning)