Statistical NLP
Winter 2011

Lecture 13: Semantic Roles / Compositional Semantics

Luke Zettlemoyer - University of Washington

[Most slides from Dan Klein]

Topics

- Today / Next Wed (Monday is a holiday!)
  - semantic role labeling (SRL)
  - inserting empty elements in parse trees
  - compositional semantics
  - learning to parse to meaning
- After that
  - discourse
  - applications

Semantic Role Labeling (SRL)

- Characterize clauses (with one verb) as *relations* with *roles*:

  [$_{\text{Judge}}$ She] **blames** [$_{\text{Evaluee}}$ the Government] [$_{\text{Reason}}$ for failing to do enough to help]

  Holman would characterise this as **blaming** [$_{\text{Evaluee}}$ the poor]

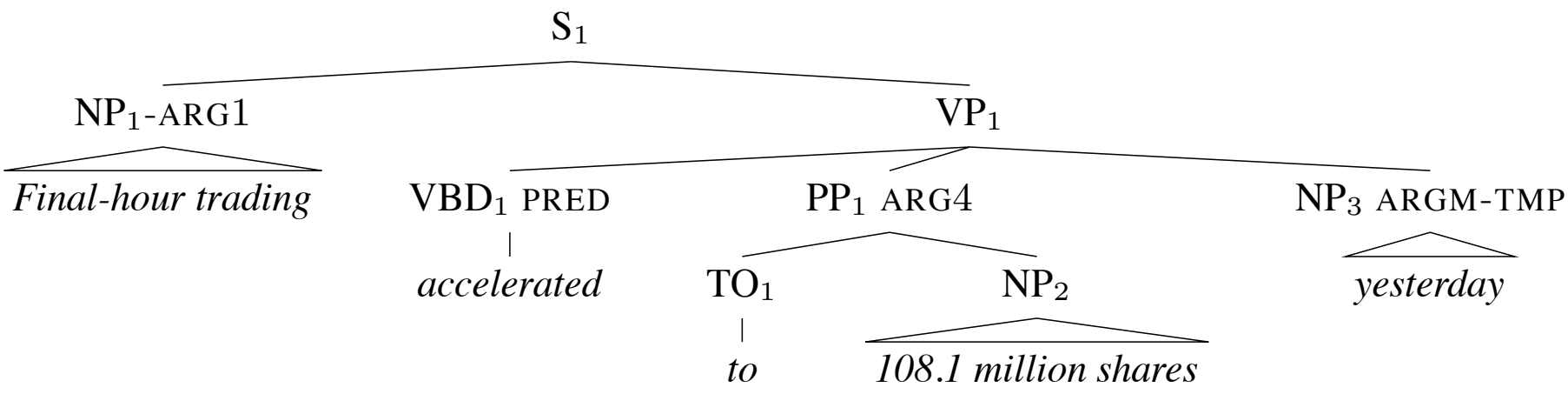
  The letter quotes Black as saying that [$_{\text{Judge}}$ white and Navajo ranchers] misrepresent their livestock losses and **blame** [$_{\text{Reason}}$ everything] [$_{\text{Evaluee}}$ on coyotes]

- Typical pipeline:
  - Parse, then label roles
  - Almost all errors locked in by parser
  - Q: Given good parses, how hard is SRL?

Thematic Roles

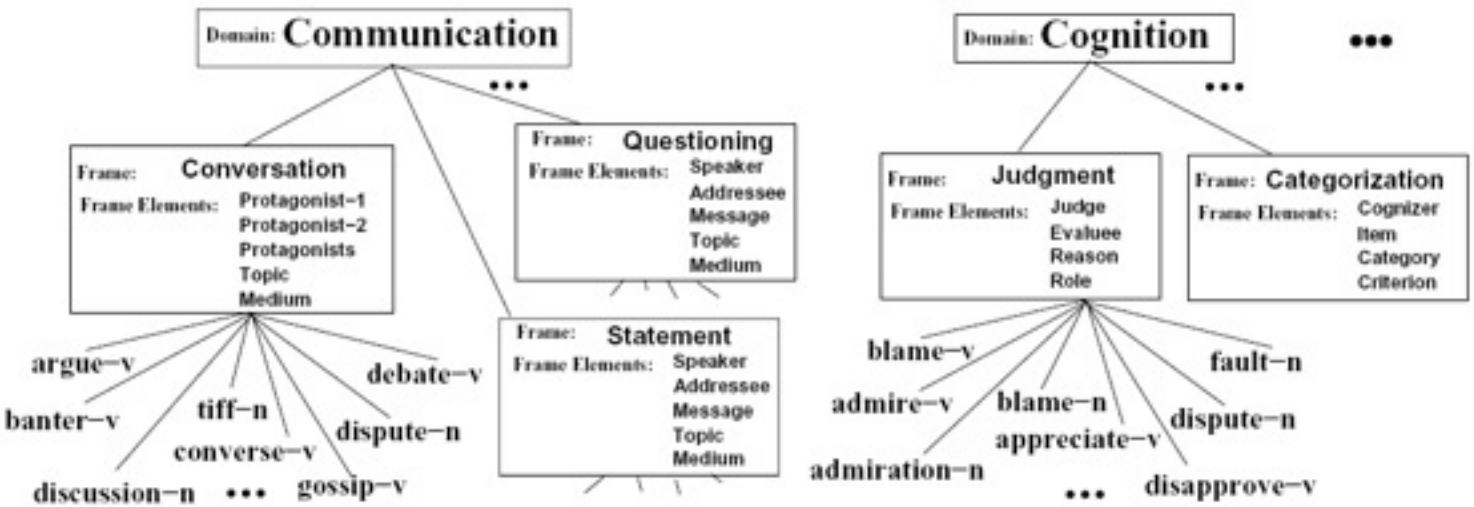
- There is a systematic relationship between syntactic positions and meaning (verb role/argument assignment):
  - Joe ate an apple.
  - An apple was eaten (by Joe).
- And, certain options do not seem to be possible
  - *An apple blahed (where blahed is a verb meaning “was eaten”)
- But, it is not always so easy
  - The sergeant played taps.
  - *Taps played.
  - Taps played quietly in the background.
  - The sergeant played a beat up old bugle.
- Relations like *subject* are syntactic, relations like *agent* or *patient* are semantic
  - Many linguistic theories use a small number (approx. 15) of semantic roles: *Agent, Patent, Instrument, Location, Time, Manner, Purpose, etc.*

SRL Example



Final-hour trading accelerated to 108.1 million shares yesterday.

PropBank / FrameNet



- FrameNet: roles shared between verbs
- PropBank: each verb has it's own roles
- PropBank more used, because it's layered over the treebank (and so has greater coverage, plus parses)

PropBank

- Core Arguments (mostly verb specific)
  - Arg0: the prototypical agent
  - Arg1: the prototypical patient or theme
  - Arg2 ... ArgN: verb specific additional arguments
- Additional arguments (ArgMs) shared across verbs
  - LOC: location, EXT: extent, DIS: discourse connectives, ADV: general-purpose, NEG: negation marker, MOD: modal verb, CAU: cause, TMP: time, PNC: purpose, MNR: manner, DIR: direction
- Frameset: verb word sense and core argument list
  - 3,300 verbs marked in data
  - 4,500 total framesets
- Three year annotation effort: 10-15 minutes to make a verb frame, except for highly ambiguous cases. 30 annotators.

PropBank Example

Sales fell to $251.2 million from $278.7 million.

fall.01
sense: *move downward*
Arg1: thing falling
Arg2: extent, distance fallen
Arg3: start point
Arg4: end point

Arg1: Sales
REL: fell
Arg2: to $251.2 million
Arg3: from $278.7 million

PropBank Example

Many of Wednesday's winners were losers yesterday as investors quickly took profits and rotated their buying to other issues, traders said.

rotate.02

sense:	*shift from one thing to another*
Arg0:	causer of shift
Arg1:	thing being changed
Arg2:	old thing
Arg3:	new thing

Arg0:	investors
Rel:	rotated
Arg1:	their buying
Arg3:	to other issues

PropBank Examples

The Central Council of Church Bell Ringers aims to improve relations with vicars

aim.01:

Sense: intend, plan
Arg0: aimer, planner
Arg1: plan, intent

Arg0: The Central Council of Church Bell Ringers
Rel: aims
Arg1: to improve relations with vicars

Banks have been aiming packages at the elderly.

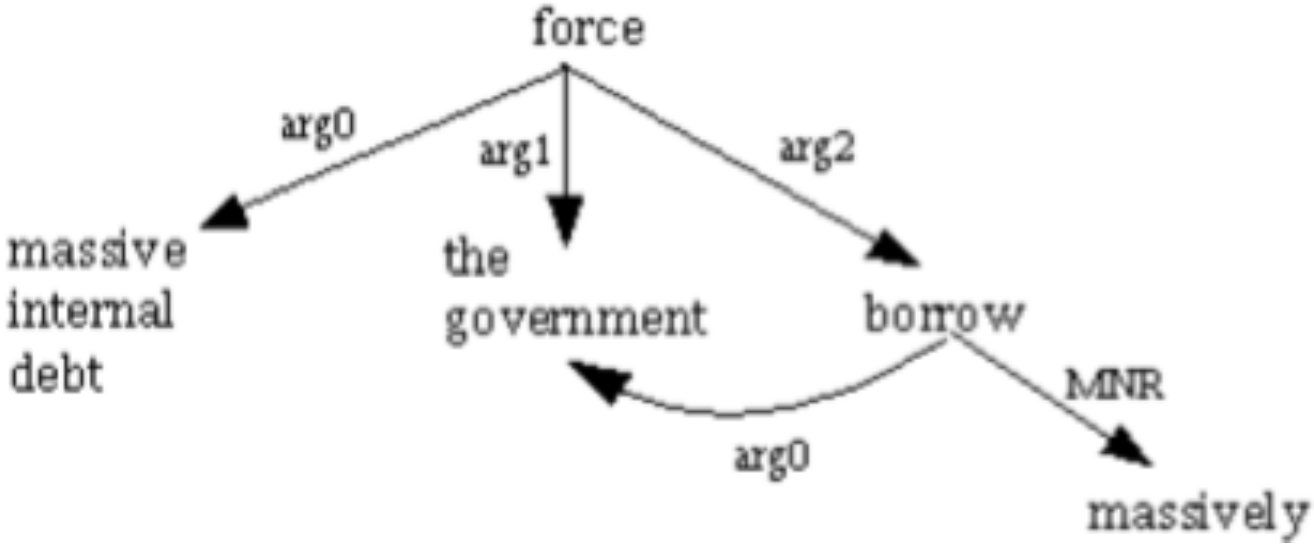
aim.02:

Sense: point (weapon) at
Arg0: aimer
Arg1: weapon, etc.
Arg2: target

Arg0: Banks
Rel: aiming
Arg1: packages
Arg2: at the elderly

Shared Arguments

massive internal debt forced the government to borrow massively



Split Constituents

By addressing those problems, Mr. Maxwell *said*, the new funds have become “extremely attractive to Japanese and other investors outside the U.S.”

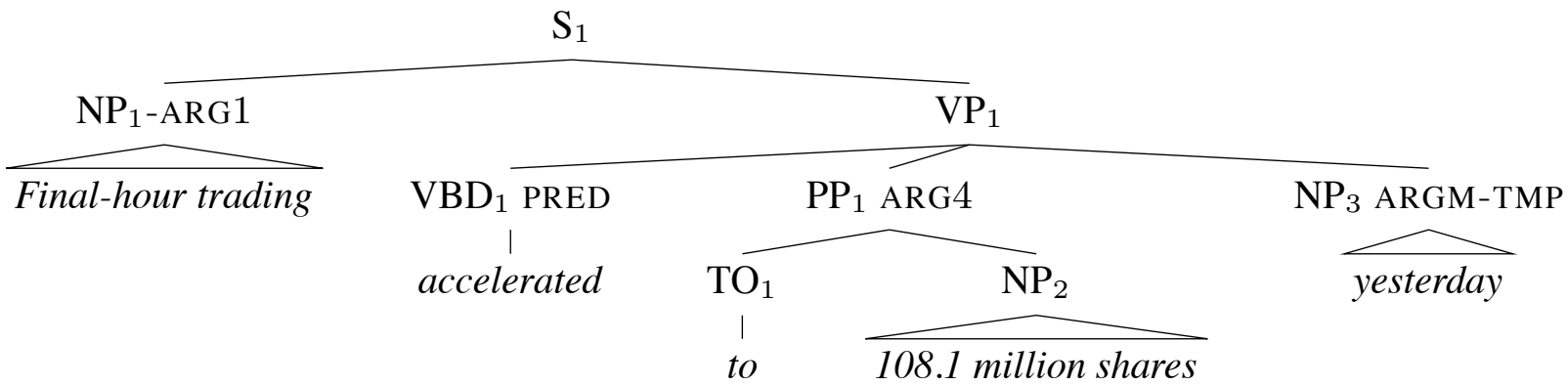
Frameset **say**

Arg0: speaker
Arg1: utterance
Arg2: listener

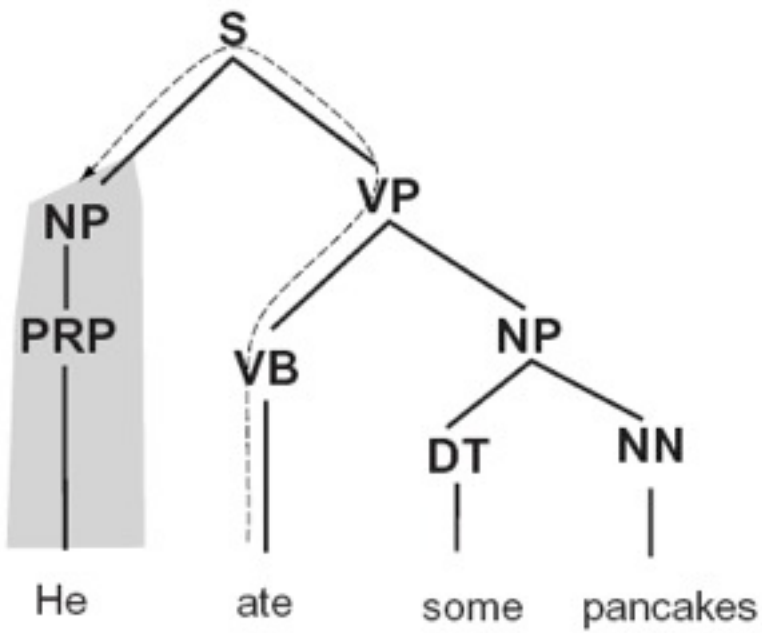
[Arg1 By addressing those problems], [Arg0 Mr. Maxwell] *said*, [Arg1 the new funds have become “extremely attractive to Japanese and other investors outside the U.S.”] (wsj 0029)

SRL: Approach

- How would you solve this problem?
  - Make use of parse trees?
  - Define features? Which ones? How do we use the trees?
  - Multi-class classification to find arguments? Independently or jointly?



Path Features



Path:	Description:
VB↑VP↓PP	PP argument / adjunct
VB↑VP↑S↓NP	subject
VB↑VP↓NP	object
VB↑VP↑VP↑S↓NP	subject (embedded VP)
VB↑VP↓ADVP	adverbial adjunct
NN↑NP↑NP↓PP	prepositional complement of noun

[Toutanova et al 04]

A Joint Model for SRL

- Problem: arguments are identified independently
- Answer: do joint classification
  - Learn local classifiers
  - Make a *n*-best list with global constraints
    - Hard: arguments do not overlap (hard to compute?)
    - Soft: features that test (potentially arbitrary) global properties
      - example: indicator for each possible sequence of arguments
  - Do re-ranking on the *n*-best list
    - multi-class classification with a log-linear model

[Toutanova et al 04]

Joint Model: Results

Oracle upper-bound with N-best:

N	CORE		ARGM	
	F1	Acc.	F1	Acc.
1	92.2	80.7	89.9	71.8
5	97.8	93.9	96.8	89.5
20	99.2	97.4	98.8	95.3
30	99.3	97.9	99.0	96.2

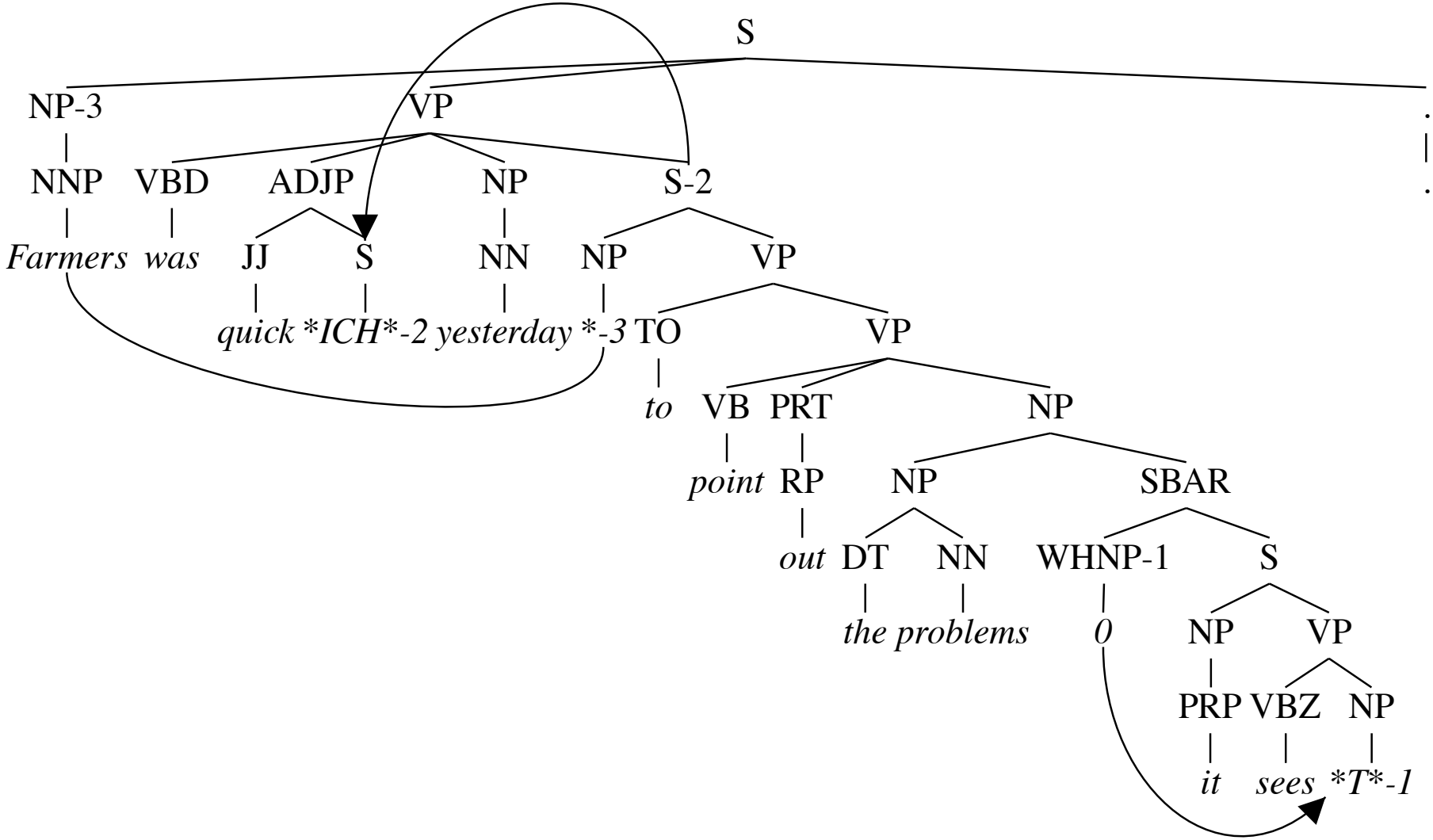
Hand-labeled parses:

Model	CORE		ARGM	
	F1	Acc.	F1	Acc.
Local	92.2	80.7	89.9	71.8
Joint	**94.7**	**88.2**	**92.1**	**79.4**

Automatic parses:

Model	CORE		ARGM	
	F1	Acc.	F1	Acc.
Local	84.1	66.5	81.4	55.6
Joint	85.8	72.7	82.9	60.8

Interaction with Empty Elements

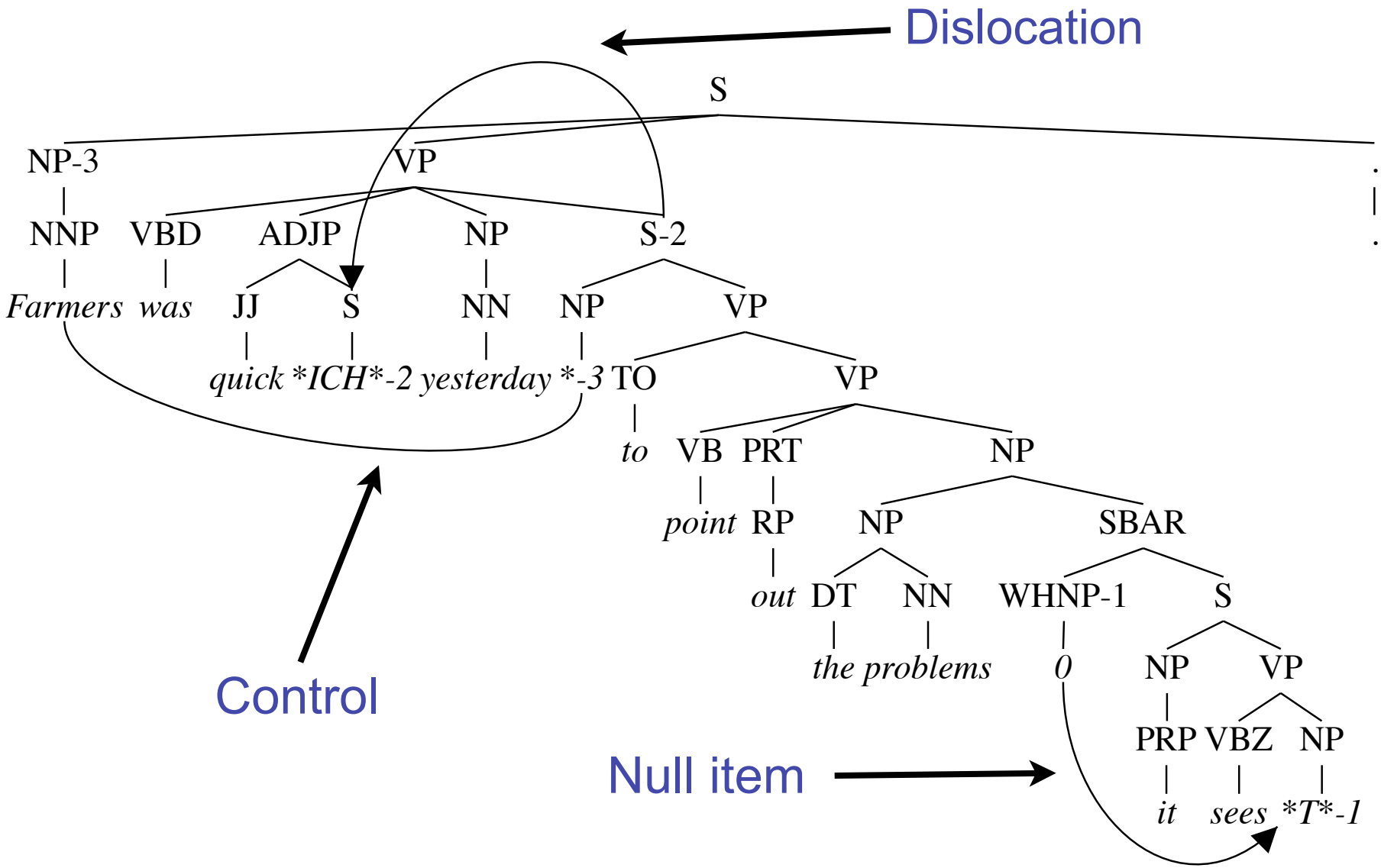


Empty Elements

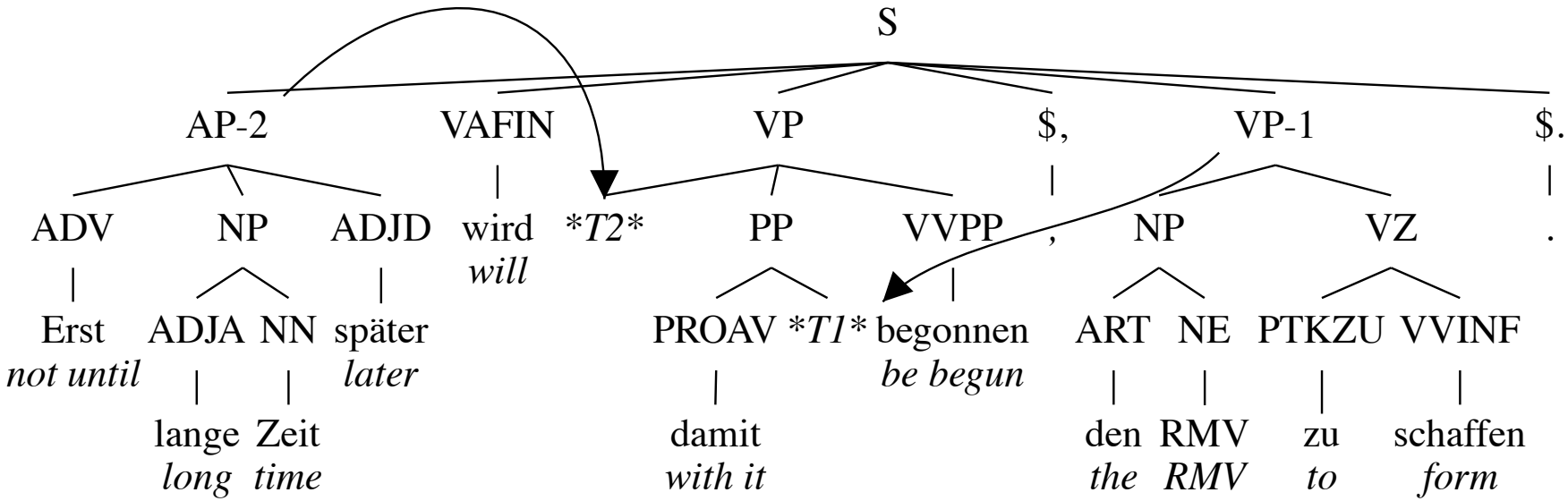
- In the PTB, three kinds of empty elements:
  - Null items (usually complementizers)
  - Dislocation (WH-traces, topicalization, relative clause and heavy NP extraposition)
  - Control (raising, passives, control, shared argumentation)

- Need to reconstruct these (and resolve any indexation)

Example: English



Example: German

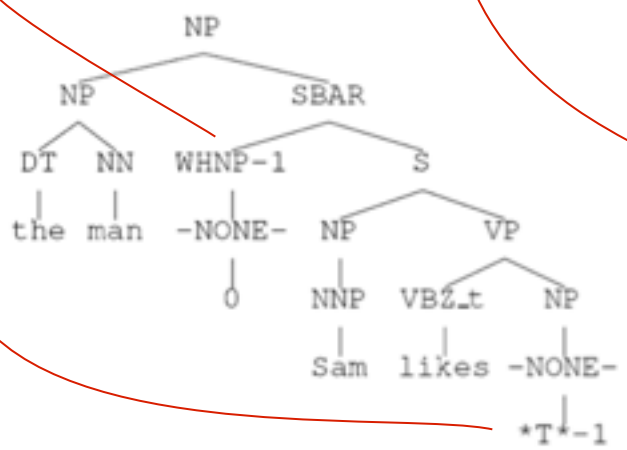


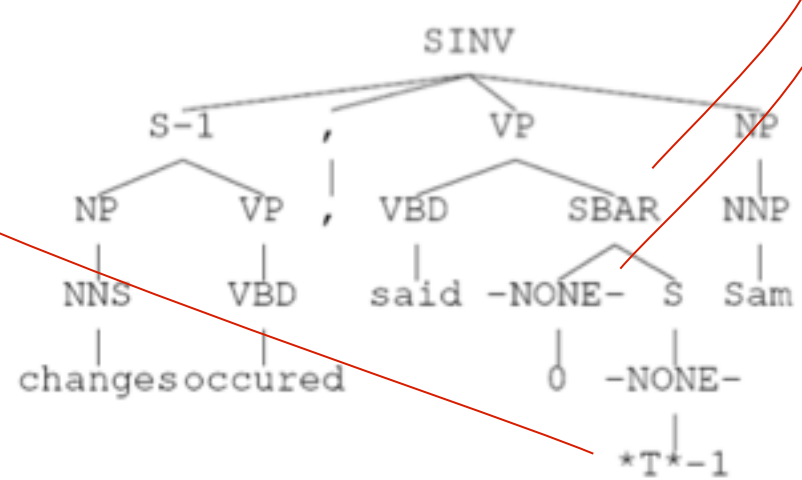
English Translation:

The RMV will not begin to be formed for a long time.

Types of Empties

Antecedent	POS	Label	Count	Description
NP	NP	*	18,334	NP trace (e.g., *Sam was seen* *)
	NP	*	9,812	NP PRO (e.g., * *to sleep is nice*)
WHNP	NP	*T*	8,620	WH trace (e.g., *the woman who you saw *T**)
		U	7,478	Empty units (e.g., *$ 25 *U**)
		0	5,635	Empty complementizers (e.g., *Sam said 0 Sasha snores*)
S	S	*T*	4,063	Moved clauses (e.g., *Sam had to go, Sasha explained *T**)
WHADVP	ADVP	*T*	2,492	WH-trace (e.g., *Sam explained how to leave *T**)
	SBAR		2,033	Empty clauses (e.g., *Sam had to go, Sasha explained (SBAR)*)
	WHNP	0	1,759	Empty relative pronouns (e.g., *the woman 0 we saw*)
	WHADVP	0	575	Empty relative pronouns (e.g., *no reason 0 to leave*)

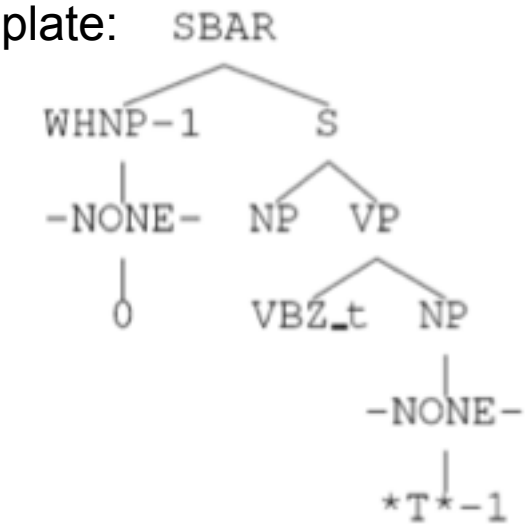


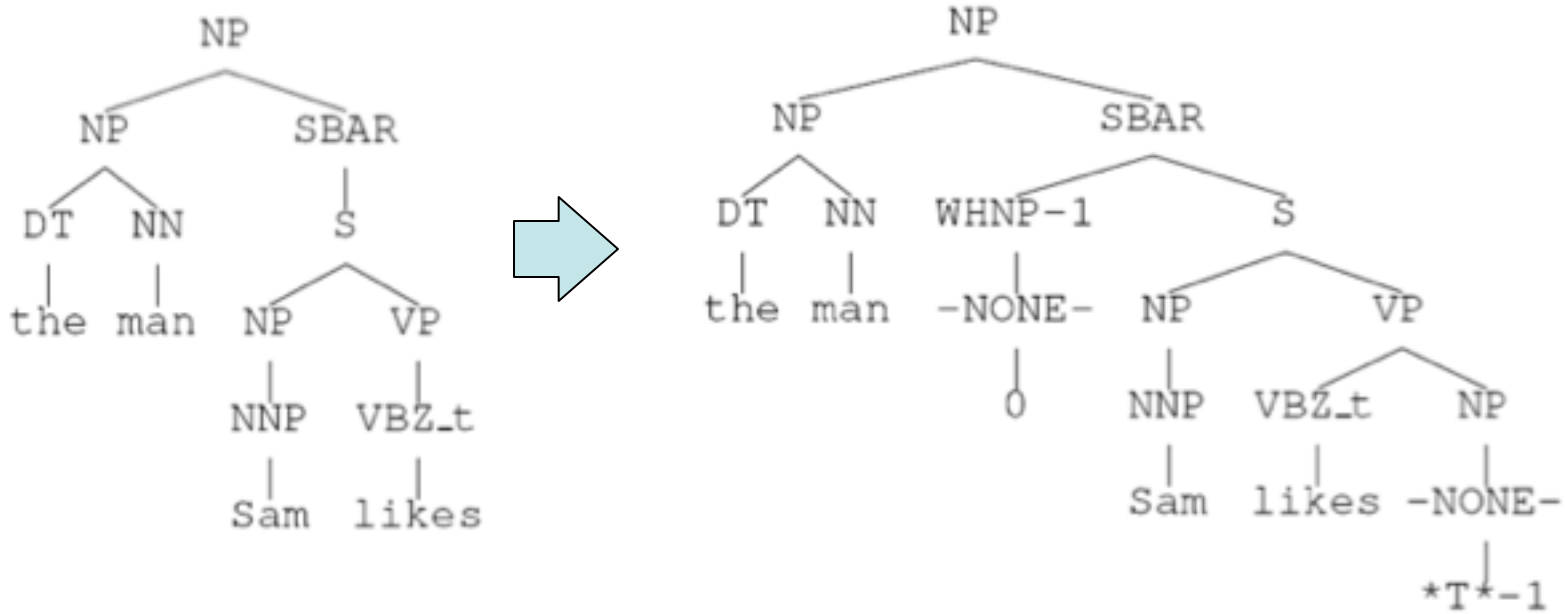


A Pattern-Matching Approach

- Problem: Treebank parsers do not mark empty elements
- Approach: build a set of rules to automatically add them to parser output

Template:





[Johnson 02]

Pattern-Matching Details

- Something like transformation-based learning
- Extract patterns
  - Details: mark transitive verbs, auxiliaries
  - Details: match legal subtrees
- Rank patterns
  - Pruning ranking: by correct / match rate on training set
  - Application priority: by depth
- Breadth first traversal (parent before children)
- Greedy match
  - favors matching large patterns first (why?)

Top Patterns Extracted

Count	Match	Pattern
5816	6223	(S (NP (-NONE- *)) VP)
5605	7895	(SBAR (-NONE- 0) S)
5312	5338	(SBAR WHNP-1 (S (NP (-NONE- *T*-1)) VP))
4434	5217	(NP QP (-NONE- *U*))
1682	1682	(NP $ CD (-NONE- *U*))
1327	1593	(VP VBN_t (NP (-NONE- *)) PP)
700	700	(ADJP QP (-NONE- *U*))
662	1219	(SBAR (WHNP-1 (-NONE- 0)) (S (NP (-NONE- *T*-1)) VP))
618	635	(S S-1 , NP (VP VBD (SBAR (-NONE- 0) (S (-NONE- *T*-1)))) .
499	512	(SINV `` S-1 , '' (VP VBZ (S (-NONE- *T*-1))) NP .)
361	369	(SINV `` S-1 , '' (VP VBD (S (-NONE- *T*-1))) NP .)
352	320	(S NP-1 (VP VBZ (S (NP (-NONE- *-1)) VP)))
346	273	(S NP-1 (VP AUX (VP VBN_t (NP (-NONE- *-1)) PP)))
322	467	(VP VBD_t (NP (-NONE- *)) PP)
269	275	(S `` S-1 , '' NP (VP VBD (S (-NONE- *T*-1))) .)

Table 2: The most common empty node patterns found in the Penn Treebank training corpus. The Count column is the number of times the pattern was found, and the Match column is an estimate of the number of times that this pattern matches some subtree in the training corpus during empty node recovery, as explained in the text.

Results

Hand-labeled parses

Parses with errors (state of the art in 2002)

Empty node		Section 23			Parser output		
POS	**Label**	P	R	f	P	R	f
(Overall)		0.93	0.83	0.88	0.85	0.74	0.79
NP	*	0.95	0.87	0.91	0.86	0.79	0.82
NP	*T*	0.93	0.88	0.91	0.85	0.77	0.81
	0	0.94	0.99	0.96	0.86	0.89	0.88
	U	0.92	0.98	0.95	0.87	0.96	0.92
S	*T*	0.98	0.83	0.90	0.97	0.81	0.88
ADVP	*T*	0.91	0.52	0.66	0.84	0.42	0.56
SBAR		0.90	0.63	0.74	0.88	0.58	0.70
WHNP	0	0.75	0.79	0.77	0.48	0.46	0.47

[Levy and Manning 04]

A Machine-Learning Approach

- Build two classifiers:
  - First one predicts where empties go
  - Second one predicts if/where they are bound
  - Use syntactic features similar to SRL (paths, categories, heads, etc)

	Performance on gold trees							Performance on parsed trees					
	ID			Rel	Combo			ID			Combo		
	P	R	F1	Acc	P	R	F1	P	R	F1	P	R	F1
WSJ(full)	92.0	82.9	87.2	95.0	89.6	80.1	84.6	34.5	47.6	40.0	17.8	24.3	20.5
WSJ(sm)	92.3	79.5	85.5	93.3	90.4	77.2	83.2	38.0	47.3	42.1	19.7	24.3	21.7
NEGRA	73.9	64.6	69.0	85.1	63.3	55.4	59.1	48.3	39.7	43.6	20.9	17.2	18.9

*F1 for individual roles, Accuracy for completely correct sentences

Semantic Interpretation

- Back to meaning!
  - A very basic approach to computational semantics
  - Truth-theoretic notion of semantics (Tarskian)
  - Assign a “meaning” to each word
  - Word meanings combine according to the parse structure
  - People can and do spend entire courses on this topic
  - We’ll spend about an hour!
- What’s NLP and what isn’t?
  - Designing meaning representations?
  - Computing those representations?
  - Reasoning with them?
- Supplemental reading will be on the web page.

Meaning

- "Meaning"
  - What is meaning?
    - "The computer in the corner."
    - "Bob likes Alice."
    - "I think I am a gummi bear."
  - Knowing whether a statement is true?
  - Knowing the conditions under which it's true?
  - Being able to react appropriately to it?
    - "Who does Bob like?"
    - "Close the door."
- A distinction:
  - Linguistic (semantic) meaning
    - "The door is open."
  - Speaker (pragmatic) meaning
- Today: assembling the semantic meaning of sentence from its parts

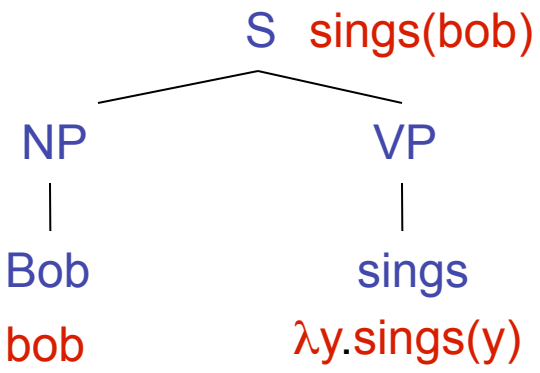
Entailment and Presupposition

- Some notions worth knowing*:
  - Entailment:
    - A entails B if A being true necessarily implies B is true
    - ? “Twitchy is a big mouse” → “Twitchy is a mouse”
    - ? “Twitchy is a big mouse” → “Twitchy is big”
    - ? “Twitchy is a big mouse” → “Twitchy is furry”
  - Presupposition:
    - A presupposes B if A is only well-defined if B is true
    - “The computer in the corner is broken” presupposes that there is a (salient) computer in the corner

*Technically, this is pragmatics

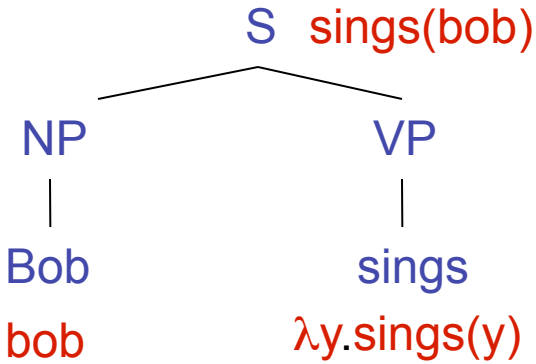
Truth-Conditional Semantics

- Linguistic expressions:
  - "Bob sings"
- Logical expressions:
  - sings(bob)
  - Could be p_1218(e_397)
- Denotation:
  - [[bob]] = some specific person (in some context)
  - [[sings(bob)]] = ???
- Types on logical expressions:
  - bob : e (for entity)
  - sings(bob) : t (for truth-value)



Truth-Conditional Semantics

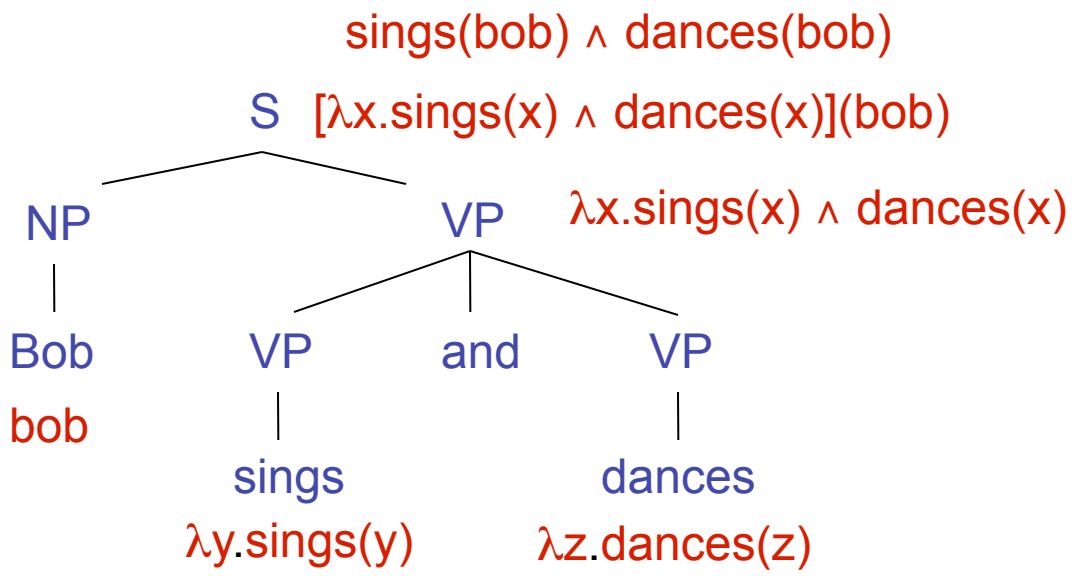
- Proper names:
  - Refer directly to some entity in the world
  - Bob : bob $[[bob]]^W \rightarrow$???
- Sentences:
  - Are either true or false (given how the world actually is)
  - Bob sings : sings(bob)



- So what about verbs (and verb phrases)?
  - sings must combine with bob to produce sings(bob)
  - The λ-calculus is a notation for functions whose arguments are not yet filled.
  - sings : λx.sings(x)
  - This is *predicate* – a function which takes an entity (type e) and produces a truth value (type t). We can write its type as e→t.
  - Adjectives?

Compositional Semantics

- So now we have meanings for the words
- How do we know how to combine words?
- Associate a combination rule with each grammar rule:
  - $S : \beta(\alpha) \rightarrow NP : \alpha \quad VP : \beta$ (function application)
  - $VP : \lambda x \,.\, \alpha(x) \wedge \beta(x) \rightarrow VP : \alpha \quad and : \varnothing \quad VP : \beta$ (intersection)
- Example:

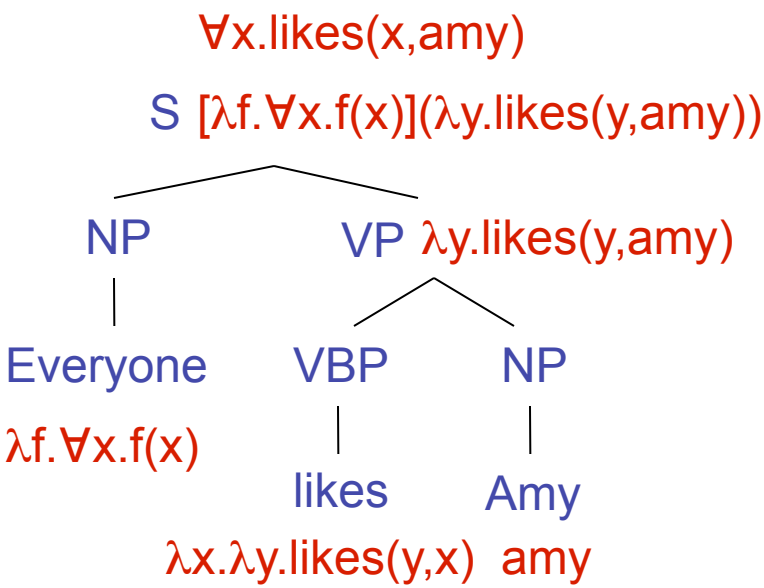


Denotation

- What do we do with logical translations?
  - Translation language (logical form) has fewer ambiguities
  - Can check truth value against a database
    - Denotation ("evaluation") calculated using the database
  - More usefully: assert truth and modify a database
  - Questions: check whether a statement in a corpus entails the (question, answer) pair:
    - "Bob sings and dances" → "Who sings?" + "Bob"
  - Chain together facts and use them for comprehension

Other Cases

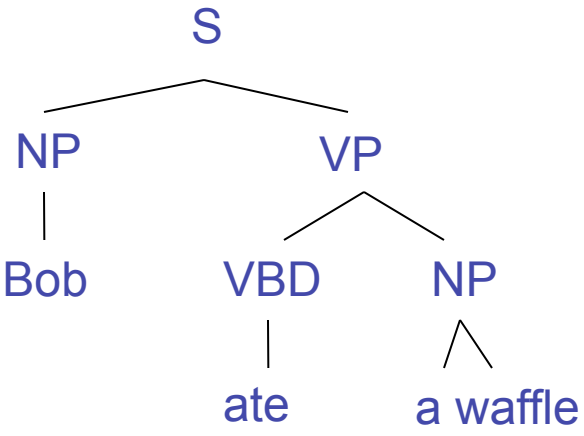
- Transitive verbs:
  - likes : $\lambda x.\lambda y.likes(y,x)$
  - Two-place predicates of type $e \rightarrow (e \rightarrow t)$.
  - likes Amy : $\lambda y.likes(y,Amy)$ is just like a one-place predicate.
- Quantifiers:
  - What does "Everyone" mean here?
  - Everyone : $\lambda f.\forall x.f(x)$
  - Mostly works, but some problems
    - Have to change our NP/VP rule.
    - Won't work for "Amy likes everyone."
  - "Everyone likes someone."
  - This gets tricky quickly!



Indefinites

- First try
  - "Bob ate a waffle" : ate(bob,waffle)
  - "Amy ate a waffle" : ate(amy,waffle)

- Can't be right!
  - $\exists$ x : waffle(x) $\wedge$ ate(bob,x)
  - What does the translation of "a" have to be?
  - What about "the"?
  - What about "every"?



Grounding

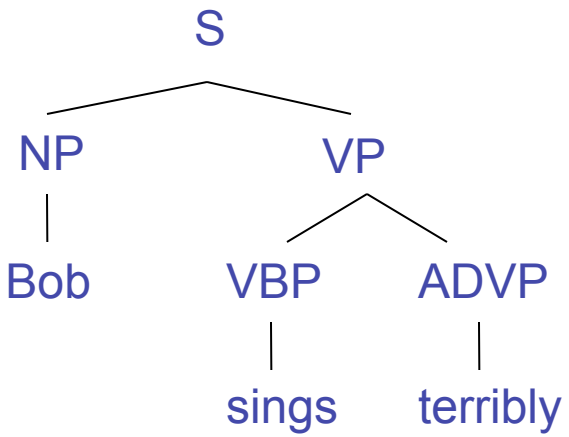
- Grounding
  - So why does the translation likes : $\lambda x.\lambda y.likes(y,x)$ have anything to do with actual liking?
  - It doesn't (unless the denotation model says so)
  - Sometimes that's enough: wire up bought to the appropriate entry in a database
- Meaning postulates
  - Insist, e.g $\forall x,y.likes(y,x) \rightarrow knows(y,x)$
  - This gets into lexical semantics issues
- Statistical version?

Tense and Events

- In general, you don't get far with verbs as predicates
- Better to have event variables e
  - "Alice danced" : danced(alice)
  - $\exists$ e : dance(e) $\wedge$ agent(e,alice) $\wedge$ (time(e) < now)
- Event variables let you talk about non-trivial tense / aspect structures
  - "Alice had been dancing when Bob sneezed"
  - $\exists$ e, e' : dance(e) $\wedge$ agent(e,alice) $\wedge$
    sneeze(e') $\wedge$ agent(e',bob) $\wedge$
    (start(e) < start(e') $\wedge$ end(e) = end(e')) $\wedge$
    (time(e') < now)

Adverbs

- What about adverbs?
  - "Bob sings terribly"
  - terribly(sings(bob))?
  - (terribly(sings))(bob)?
  - $\exists e$ present(e) $\wedge$ type (e, singing) $\wedge$ agent (e,bob) $\wedge$ manner(e, terrible) ?
  - It's really not this simple..



Propositional Attitudes

- "Bob thinks that I am a gummi bear"
  - thinks(bob, gummi(me)) ?
  - thinks(bob, "I am a gummi bear") ?
  - thinks(bob, ^gummi(me)) ?
- Usual solution involves intensions (^X) which are, roughly, the set of possible worlds (or conditions) in which X is true
- Hard to deal with computationally
  - Modeling other agents models, etc
  - Can come up in simple dialog scenarios, e.g., if you want to talk about what your bill claims you bought vs. what you actually bought

Trickier Stuff

- Non-Intersective Adjectives
  - green ball : $\lambda x.[green(x) \wedge ball(x)]$
  - fake diamond : $\lambda x.[fake(x) \wedge diamond(x)]$? $\longrightarrow$ $\lambda x.[fake(diamond(x))$
- Generalized Quantifiers
  - the : $\lambda f.[\textit{unique-member}(f)]$
  - all : $\lambda f.\ \lambda g\ [\forall x.f(x) \rightarrow g(x)]$
  - most?
  - Could do with more general second order predicates, too (why worse?)
    - the(cat, meows), all(cat, meows)
- Generics
  - "Cats like naps"
  - "The players scored a goal"
- Pronouns (and bound anaphora)
  - "If you have a dime, put <u>it</u> in the meter."

- … the list goes on and on!

Multiple Quantifiers

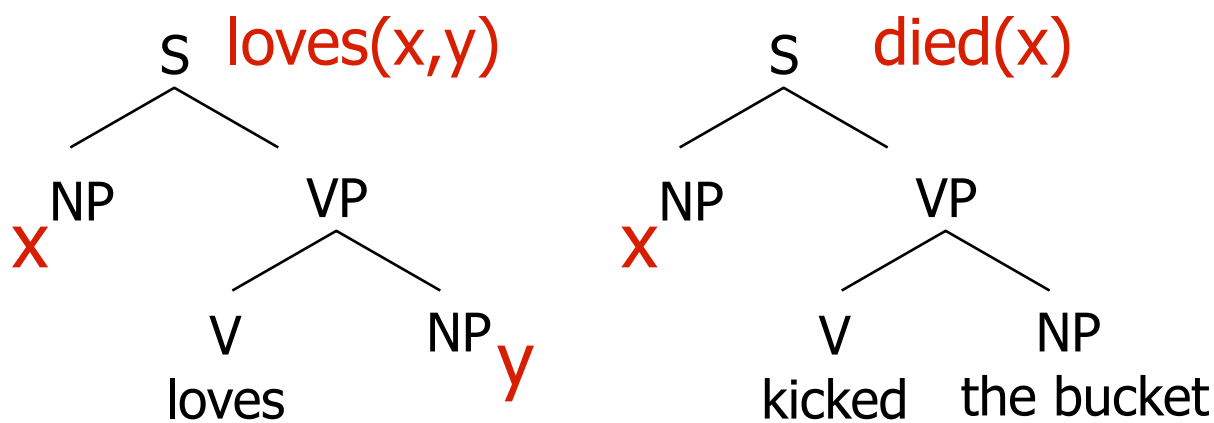
- Quantifier scope
  - Groucho Marx celebrates quantifier order ambiguity:

    "In this country <u>a woman</u> gives birth <u>every 15 min</u>.
    Our job is to find that woman and stop her."

- Deciding between readings
  - "Bob bought a pumpkin every Halloween"
  - "Bob put a warning in every window"
  - Multiple ways to work this out
    - Make it syntactic (movement)
    - Make it lexical (type-shifting)

Implementation, TAG, Idioms

- Add a "sem" feature to each context-free rule
  - S → NP loves NP
  - S[sem=loves(x,y)] → NP[sem=x] loves NP[sem=y]
  - Meaning of S depends on meaning of NPs
- TAG version:



- Template filling: S[sem=showflights(x,y)] →
  I want a flight from NP[sem=x] to NP[sem=y]

Modeling Uncertainty

- Gaping hole warning!
- Big difference between statistical disambiguation and statistical reasoning.

  The scout saw the enemy soldiers with night goggles.

  - With probabilistic parsers, can say things like “72% belief that the PP attaches to the NP.”
  - That means that *probably* the enemy has night vision goggles.
  - However, you can’t throw a logical assertion into a theorem prover with 72% confidence.
  - Not clear humans really extract and process logical statements symbolically anyway.
  - Use this to decide the expected utility of calling reinforcements?

- In short, we need probabilistic reasoning, not just probabilistic disambiguation followed by symbolic reasoning!