CS114 Lecture 19

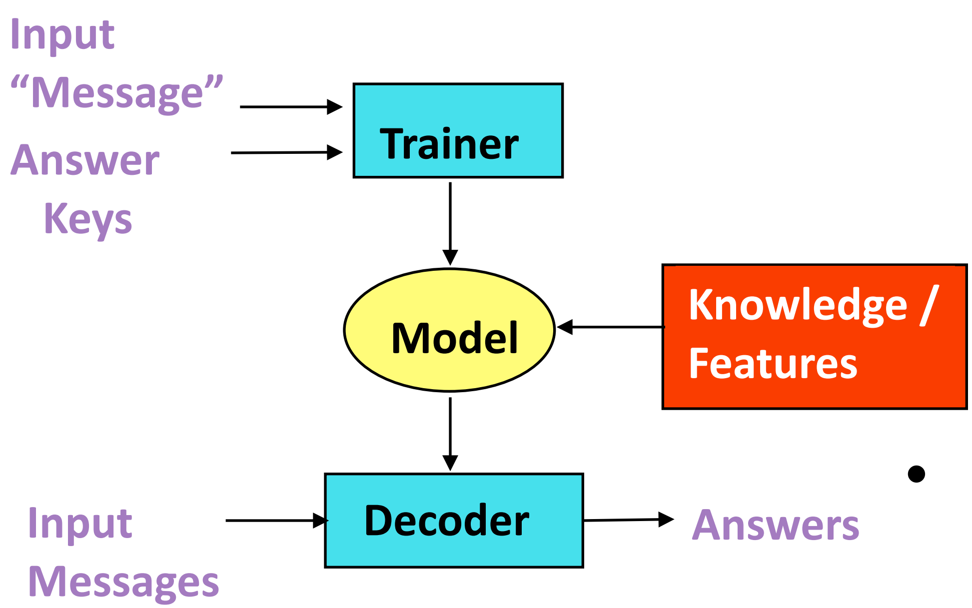
Entropy and Features

April 7, 2014

Professor Meteer

Thanks for Jurafsky & Martin & Prof. Pustejovksy for slides

Speech and NL Paradigm



- **Requirements:**
  - Annotation of messages with keys
  - Features: the Linguistic and Domain Knowledge
  - Statistical Model
  - Training Algorithm
  - Decoding Algorithm
- **Benefits:**
  - Statistical model can combine multiple kinds of information
  - Degrades "softly", finding the most likely answer
  - Learns what information is important to make a decision

Supervised Learning for Language Technologies

Technology	Input	Answers
Speech Recognition	Audio	Transcription
Optical Character Recognition	Image	Characters
Topic classification	Document	Topic labels
Information retrieval	Query	Document
Named entity extraction	Text or speech	Names and categories

Advantages of the Learning Approach

- Large amounts of electronic text are now available.
- Annotating corpora is easier and requires less expertise than manual knowledge engineering.
- Learning algorithms have progressed to be able to handle large amounts of data and produce accurate probabilistic knowledge.
- The probabilistic knowledge acquired allows robust processing that handles linguistic regularities as well as exceptions.

The Cycle of Computational Linguistics

- We can study anything about language ...

1. Formalize some insights
2. Study the formalism mathematically
3. Develop & implement algorithms

   Select the features!

4. Test on real data

Feature types

- Target
  - What you are trying to learn
  - Consider complexity
    - 43 parts of speech or 118?
- "Features"
  - Selected knowledge that is used to train the model
  - Must be something I can measure/count!
  - Some are more obvious than others

Which features to use?

Most crucial decision you'll make!

1. Topic
   - Words, phrases, ?
2. Author
   - Stylistic features
3. Sentiment
   - Adjectives, ?
4. Spam
   - Specialized vocabulary

How to choose features

- Consider cost
  - Words vs. POS vs parse tree
- Observable/countable
- Differentiating
  - Remove “non-informative” terms from documents
- Questions to consider
  - Stemmed or surface form?
  - Single words or phrases?
  - Words or word classes?

A Simple Example

- Gender identification based on names
  - Hypothesis
    - Names ending with a, e, and i are likely to be female
    - Names ending with k, o, r, s, and t are likely to be male
- Build a classifier
  - Use marked data, divide training and test sets
- Analyze errors:
  - Female -> male: Cindelyn, Kathryn
  - Male -> female: Rich, Mitch
- Adjust features
  - Not just last letter, could be last two letters
- Repeat

Speech recognition

- Acoustic signal -> accurate text transcription
- "Features" are
  - the phonetic spellings of the words
  - And the "context"
    - Neighboring phonemes
    - Previous words
- The more words, the more data you need
  - Should you stem the words?
  - Should you combine them into multiwords?

Part of Speech Tagging

- “Closed set” for known words
  - Dictionary of words and possible tags
  - Data marked with tags to determine “Word emit” probability and context (n-gram)
- How many tags? Is more better? Worse?
- How big a context window? 3-gram? 7-gram?
- Feature set for unknown words
  - Inflectional endings (-ed, -s, -ing)
  - Derivational endings (-ion, -ly, -ive, …)
  - Hyphenation (+-)
  - Capitalization (4 values: +-capital, +-initial)
- Why these?

Probabilistic CFG

- Simplest:
  - Features are the rules and rule context
- How general/expressive should the rules be?
- Problems
  - Independence assumptions misses structural dependencies
    - E.g pronouns more likely in subject position
    - Solution: More nonterminals, eg NP-SUB
    - But this is just additional features
  - Lack of lexical sensitivity
    - Make the head word a feature of the rule
    - Now how many rules?

Word Sense Disambiguation

- Supervised machine learning approach:
  - A training corpus of words tagged in context with their sense
  - Corpus is used to train a classifier that can tag words in new text
- Summary of what we need:
  - the **tag set** (“sense inventory”)
  - the **training corpus**
  - A set of **features** extracted from the training corpus
  - A **classifier**

Feature vectors

- A simple representation for each observation (each instance of a target word)
  - Vectors of sets of feature/value pairs
    - I.e. files of comma-separated values
  - These vectors should represent the window of words around the target

Collocational

- Position-specific information about the words in the window
- guitar and bass player stand
  - [guitar, NN, and, CC, player, NN, stand, VB]
  - Word_{n-2}, POS_{n-2}, word_{n-1}, POS_{n-1}, Word_{n+1} POS_{n+1}…
  - In other words, a vector consisting of
  - [position n word, position n part-of-speech…]

Word Similarity: Context vector

- Consider a target word *w*
- Suppose we had one binary feature f_i for each of the *N* words in the lexicon v_i
- Which means "word v_i occurs in the neighborhood of *w*"
- w=(f1,f2,f3,…,fN)
- If w=tezguino, v1 = bottle, v2 = drunk, v3 = matrix:
- w = (1,1,0,…)

Co-occurrence vectors based on dependencies

- For the word "cell": vector of NxR features
  – R is the number of dependency relations
- What do I need for this?

	subj-of, absorb	subj-of, adapt	subj-of, behave	...	pobj-of, inside	pobj-of, into	...	nmod-of, abnormality	nmod-of, anemia	nmod-of, architecture	...	obj-of, attack	obj-of, call	obj-of, come from	obj-of, decorate	...	nmod, bacteria	nmod, body	nmod, bone marrow
cell	1	1	1		16	30		3	8	1		6	11	3	2		3	2	2

Semantic Role Labeling

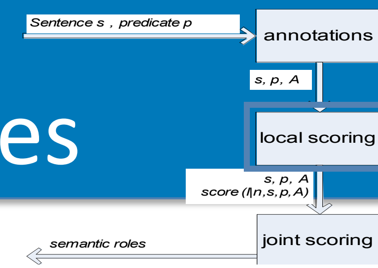
- What's the target? What am I trying to learn?
  - Traditional thematic roles
    - Agent, patient, theme, goal, instrument
  - FrameNet
    - Seller, buyer
  - "Agnostic" Propbank
    - A0, A1, A2
- What features are available that would help to model the distinctions?

Steps in SRL

From Xue & Palmer EMLNP 2004

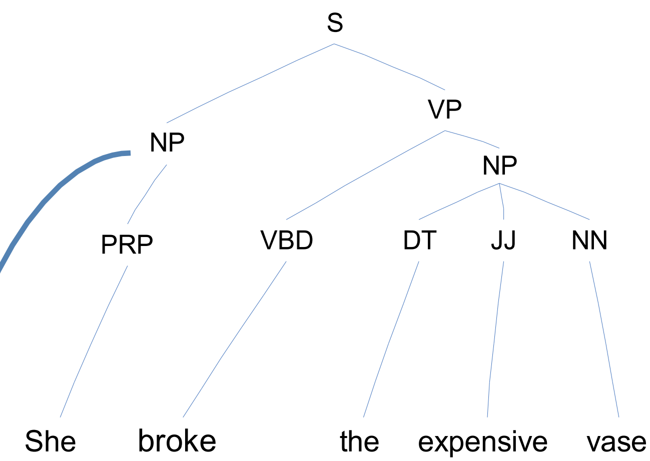
- Stage 1: Filter out constituents that are clearly not semantic arguments to the predicate in question (saves time)
- Stage 2: Classify the candidates derived from the first stage as either semantic arguments or non-arguments.
- Stage 3: Run a multi-category classifier to classify the constituents that are labeled as arguments into one of the classes plus NULL.

Gildea & Jurafsky (2002) Features



- Key early work
  - Future systems use these features as a baseline

- Constituent Independent
  - Target predicate (lemma)
  - Voice
  - Subcategorization
- Constituent Specific
  - Path
  - Position (*left*, *right*)
  - Phrase Type
  - Governing Category (*S* or *VP*)
  - Head Word

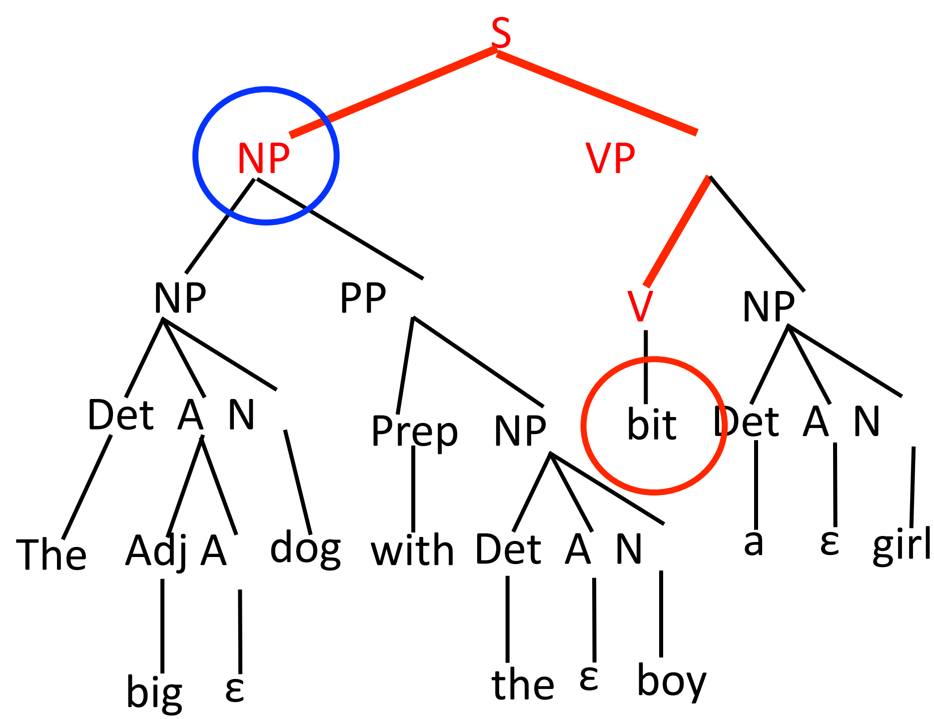


Target	*broke*
Voice	*active*
Subcategorization	*VP→VBD NP*
Path	*VBD↑VP↑S↓NP*
Position	*left*
Phrase Type	*NP*
Gov Cat	*S*
Head Word	*She*

Parse Tree Path Feature: Example 1

Path Feature Value:

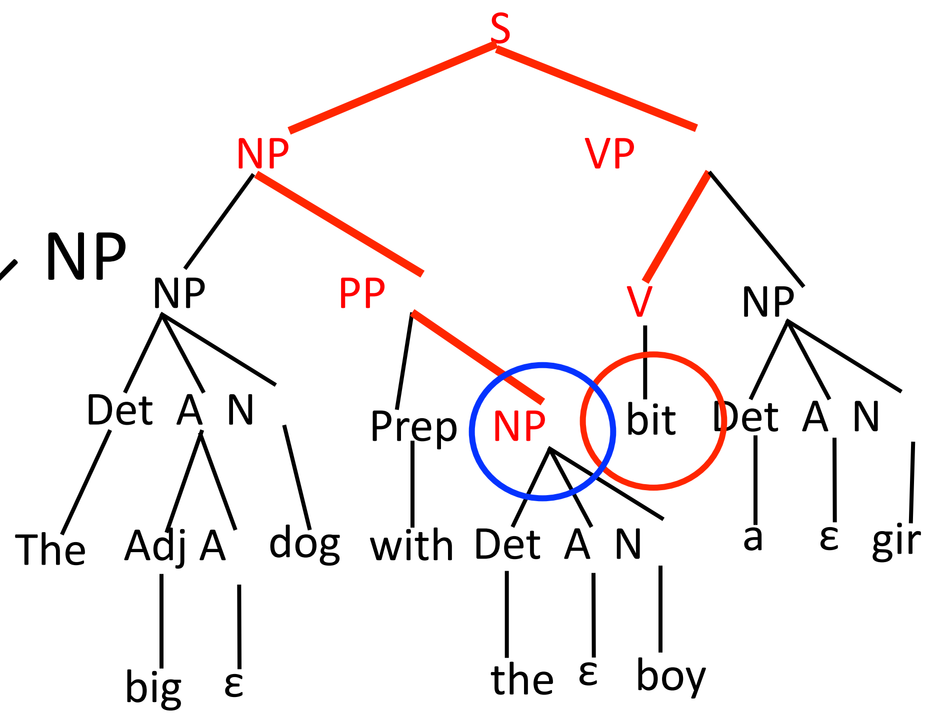
$$V \uparrow VP \uparrow S \downarrow NP$$



Parse Tree Path Feature: Example 2

Path Feature Value:

V ↑ VP ↑ S ↓ NP ↓ PP ↓ NP



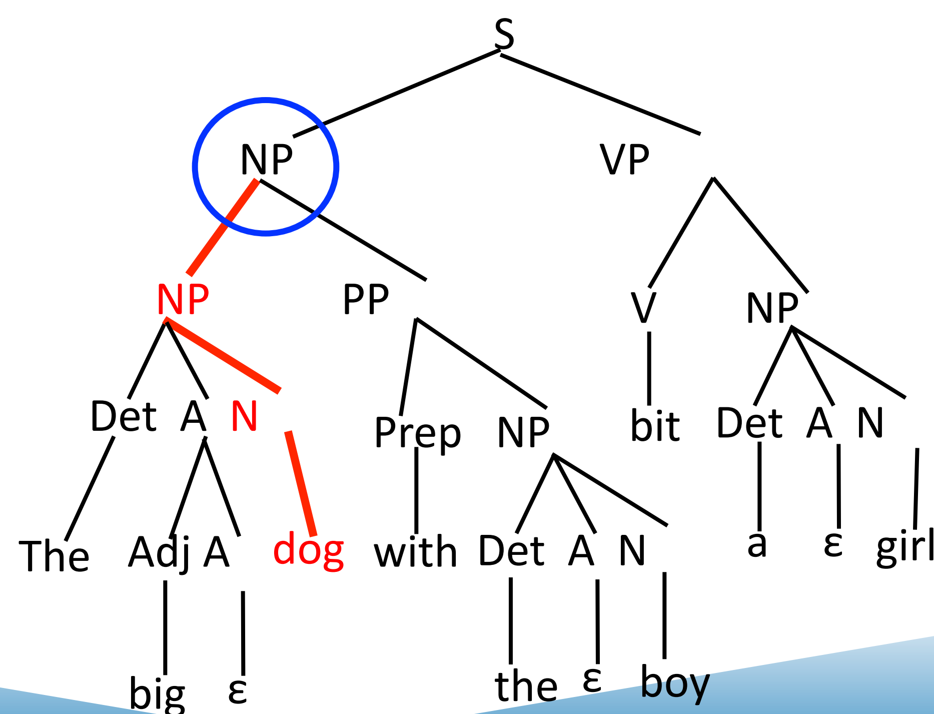
From Randy Mooney, UT Austin

Head Word Feature Example

- There are standard syntactic rules for determining which word in a phrase is the **head**.

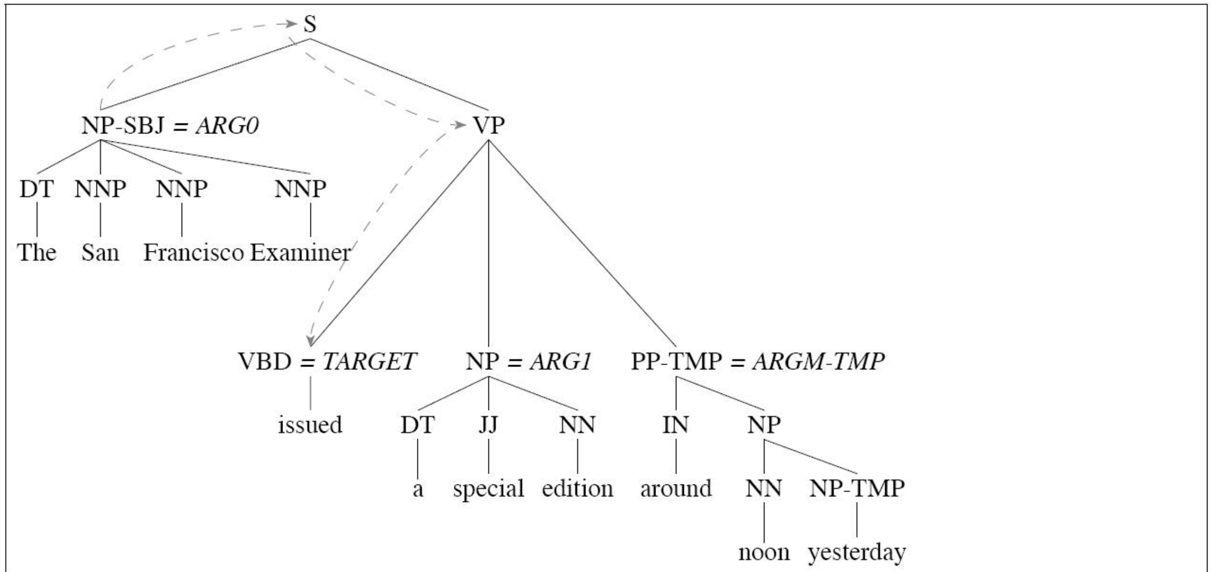
Head Word:

dog



From Randy Mooney, UT Austin

Another example



Target	*issued*
Voice	*active*
Subcategorization	*VP→VBD NP PP*
Path	*VBD↑VP↑S↓NP*
Position	*left*
Phrase Type	*NP*
Gov Cat	*S*
Head Word	*Examiner*

Target	*issued*
Voice	*active*
Subcategorization	*VP→VBD NP PP*
Path	*VBD↑VP↓NP*
Position	*right*
Phrase Type	*NP*
Gov Cat	*VP*
Head Word	*edition*

Summary "Standard" features

- Predicate The predicate itself.
- Path The minimal path from the constituent being classified to the predicate.
- Phrase Type The syntactic category (NP, PP, etc.) of the constituent being classified.
- Position The relative position of the constituent being classified with regard to the predicate (before or after)
- Voice Whether the predicate is active or passive.
- Head Word The head word of the constituent being classified.
- Sub-categorization The phrase structure rule expanding the parent of the predicate.

Argument Identification

- A subset of features and their combination contribute most to argument identification
  - path,
  - head word, head word part-of-speech,
  - predicate - phrase type combination,
  - predicate- head word combination,
  - distance between constituent and predicate, with the predicate specified.

Argument identification

- Some features to not help discriminate argument identification
  - path: Can't distinguish between sisters
    - Direct object & indirect object not distinct
  - Subcategorization: Shared by all of the arguments
  - Voice: Same for all args, mabey combine with arg/label
  - phrase type: Does help but would be stronger if pared with the predicate
  - head word: Also should be pared with predicate

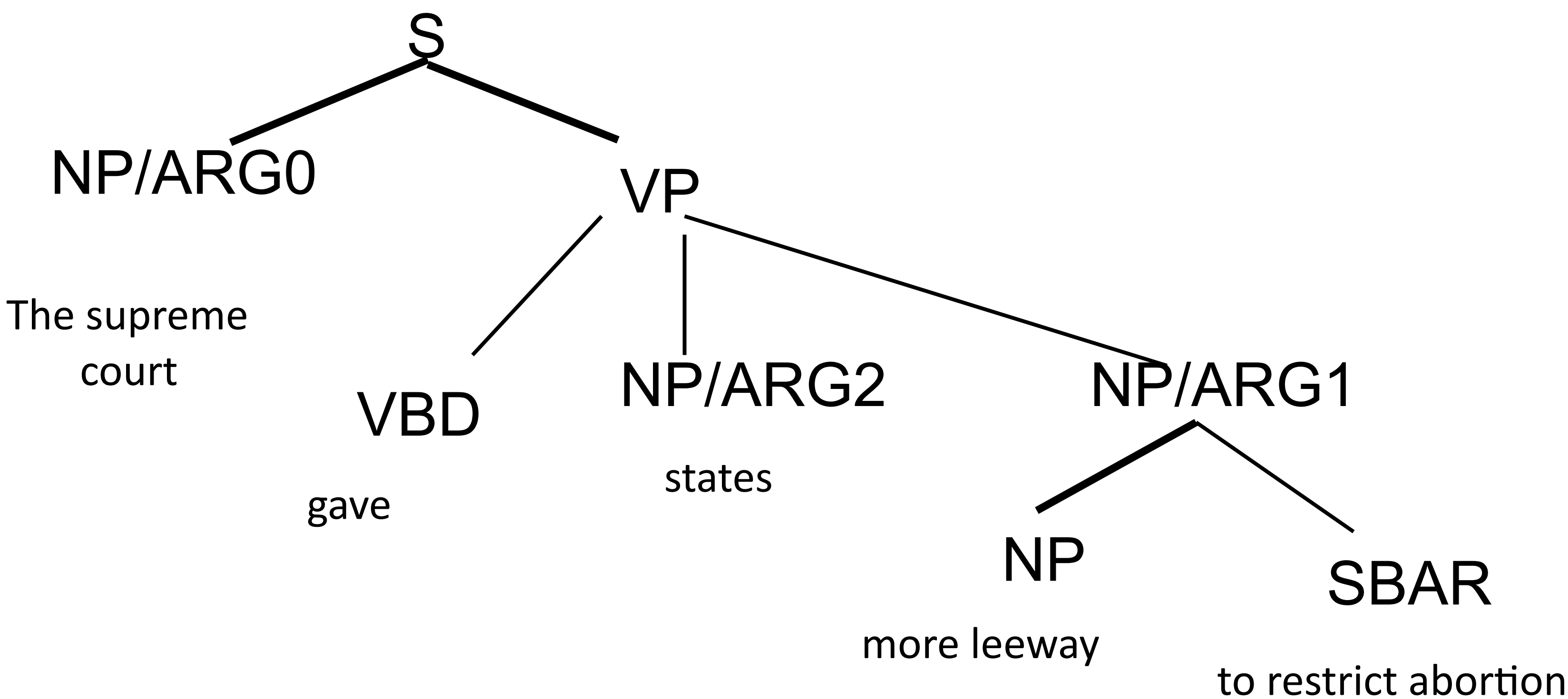
New features for Argument Identification

- Syntactic frame: varies with the constituent being classified to complement the path and subcat features
- Lexicalized constituent type: combination of the predicate lemma and the phrase type, rather than the phrase type itself, e.g. give np.
- Lexicalized head : predicate lemma and the head word combination as a feature, e.g. give states.
- Voice position combination: voice position combination as a feature, e.g. passive before.
- Head of PP: parent If the parent of the current constituent is a PP, then the head of this PP, the preposition is also used as a feature.

Performance per feature

Features	Accuracy	Gold(f)
Baseline	88.09	82.89
Syntactic frame	89.82	84.64
Pred-Head	88.69	83.77
Pred-POS	89.12	83.81
Voice position	88.44	82.57
PP parent	89.53	84.34
First word	88.60	83.01
Last word	88.64	83.51
Left sister	89.20	83.74
all	92.95	88.51

Syntactic Frames



Syntactic frame for "states": np_give_NP_np

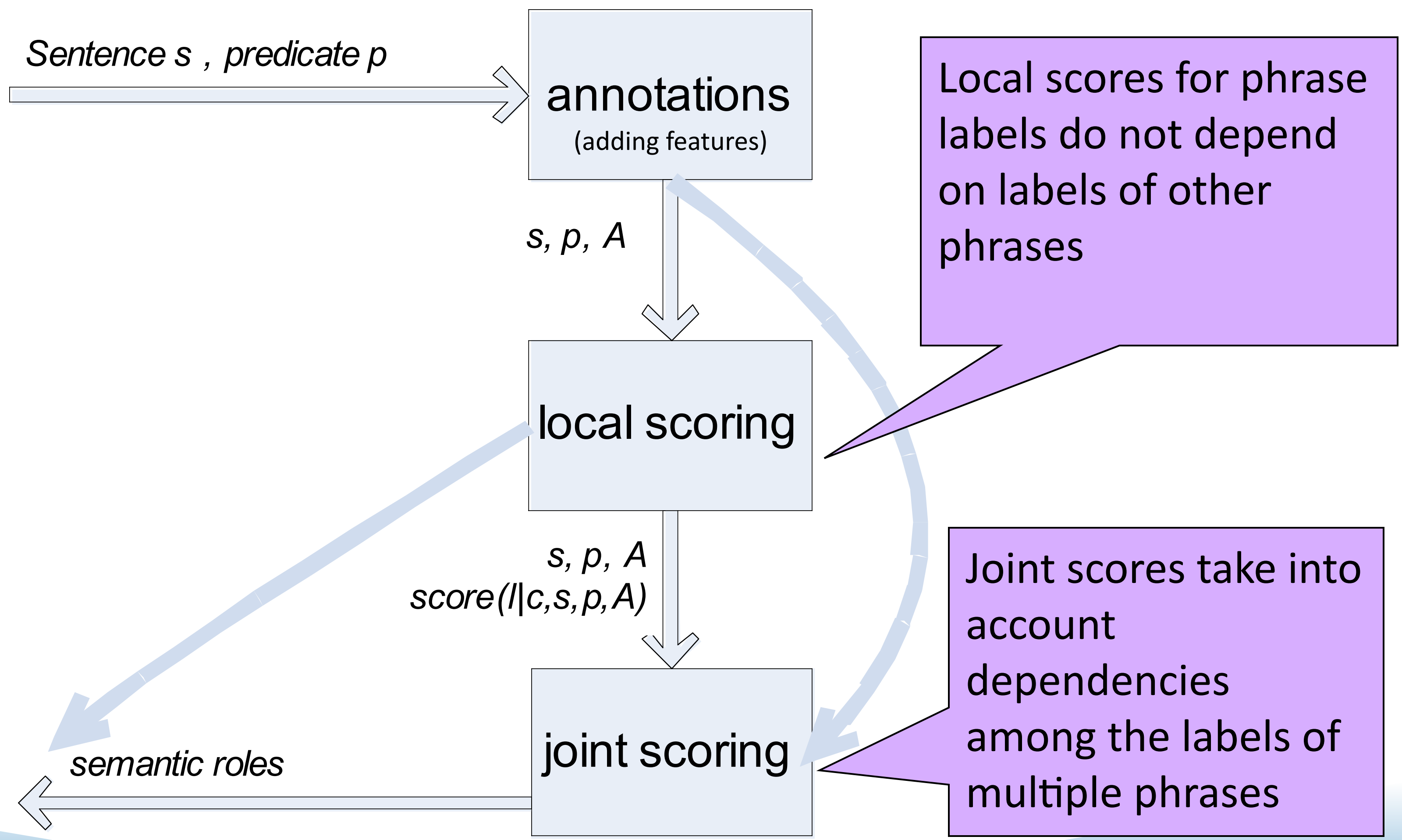
Syntactic from for "more leeway…": np_give_np_NP

Pradhan et al. 2004 features

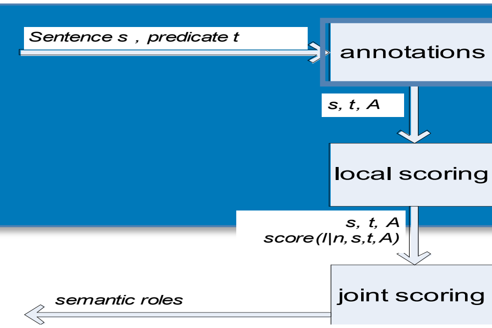
- Predicate cluster
- Noun head and POS of PP constituent
- Verb sense
- Partial path
- Named entities in constituent (7) [Surdeanu et al., 2003]
- Head word POS [Surdeanu et al., 2003]
- First and last word in constituent and their POS
- Parent and sibling features
- Constituent tree distance
- Ordinal constituent position
- Temporal cue words in constituent
- Previous 2 classifications

Slide from Sameer Pradhan

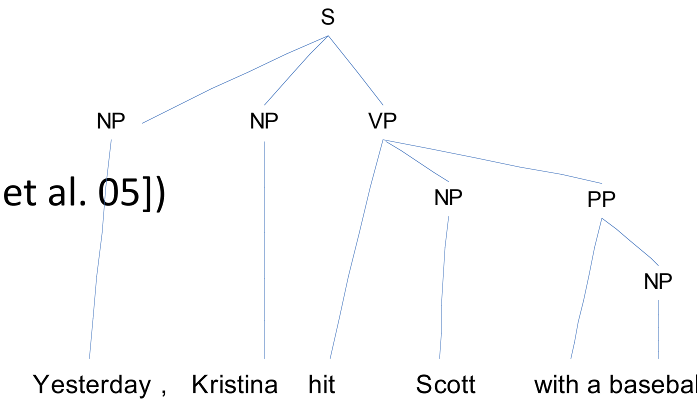
Basic Architecture of a Generic SRL System



Annotations Used



- Syntactic Parsers
  - Collins', Charniak's (most systems)
  - CCG parses ([Gildea & Hockenmaier 03],[Pradhan et al. 05])
  - TAG parses ([Chen & Rambow 03])



- Shallow parsers

  [$_{NP}$Yesterday] , [$_{NP}$Kristina] [$_{VP}$hit] [$_{NP}$Scott] [$_{PP}$with] [$_{NP}$a baseball].

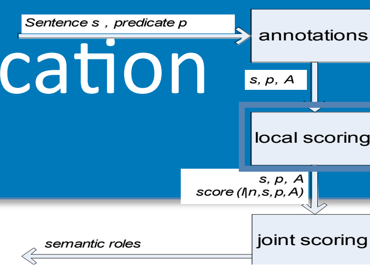
- Semantic ontologies (WordNet, automatically derived), and named entity classes

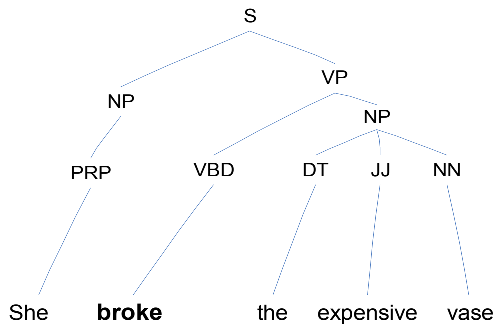
  (v) **hit** (cause to move by striking)

  WordNet hypernym

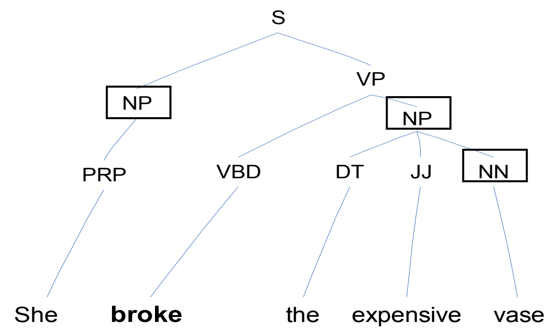
  propel, impel *(*cause to move forward with force*)*

Combining Identification and Classification Models

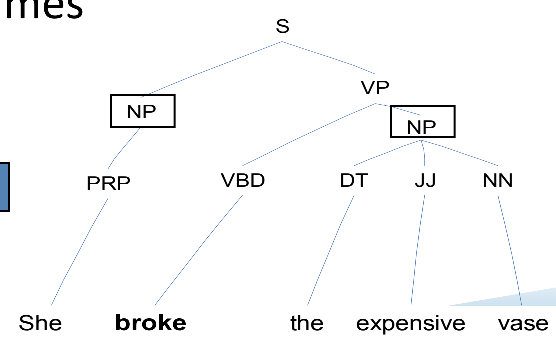




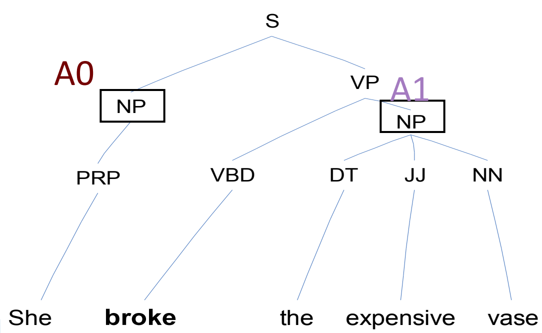
Step 1. *Pruning.* Using a hand-specified filter.



Step 2. *Identification.* Identification model (filters out candidates with high probability of NONE)

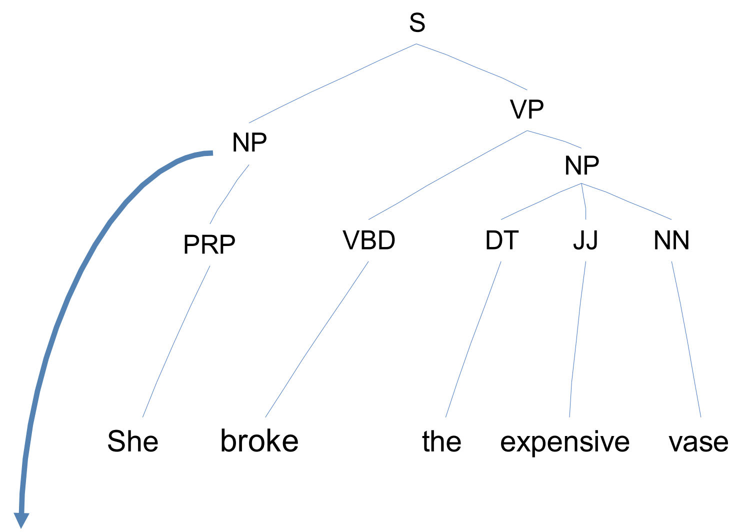


Step 3. *Classification.* Classification model assigns one of the argument labels to selected nodes (or sometimes possibly NONE)



Gildea & Jurafsky (2002) Features

- Key early work
  - Future systems use these features as a baseline
- Constituent Independent
  - Target predicate (lemma)
  - Voice
  - Subcategorization
- Constituent Specific
  - Path
  - Position (*left, right*)
  - Phrase Type
  - Governing Category (*S* or *VP*)
  - Head Word

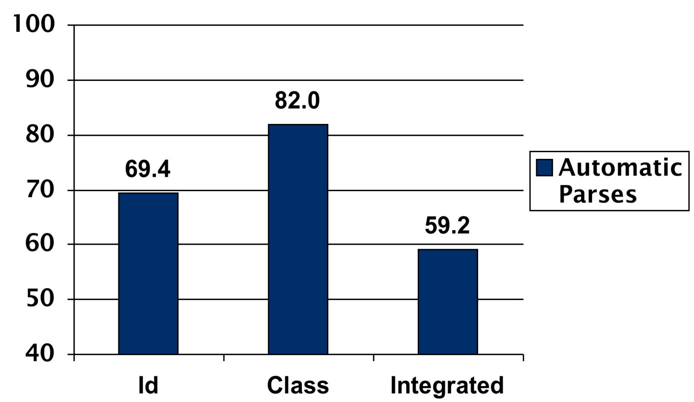


Target	*broke*
Voice	*active*
Subcategorization	*VP→VBD NP*
Path	*VBD↑VP↑S↓NP*
Position	*left*
Phrase Type	*NP*
Gov Cat	*S*
Head Word	*She*

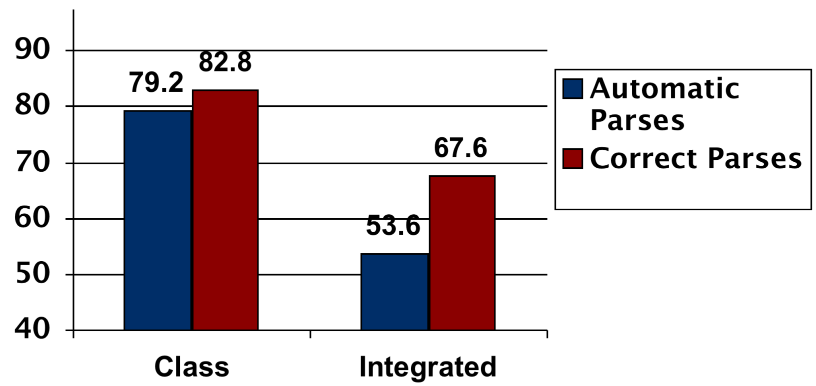
Performance with Baseline Features using the G&J Model

- **Machine learning algorithm:** interpolation of relative frequency estimates based on subsets of the 7 features introduced earlier

FrameNet Results



Propbank Results

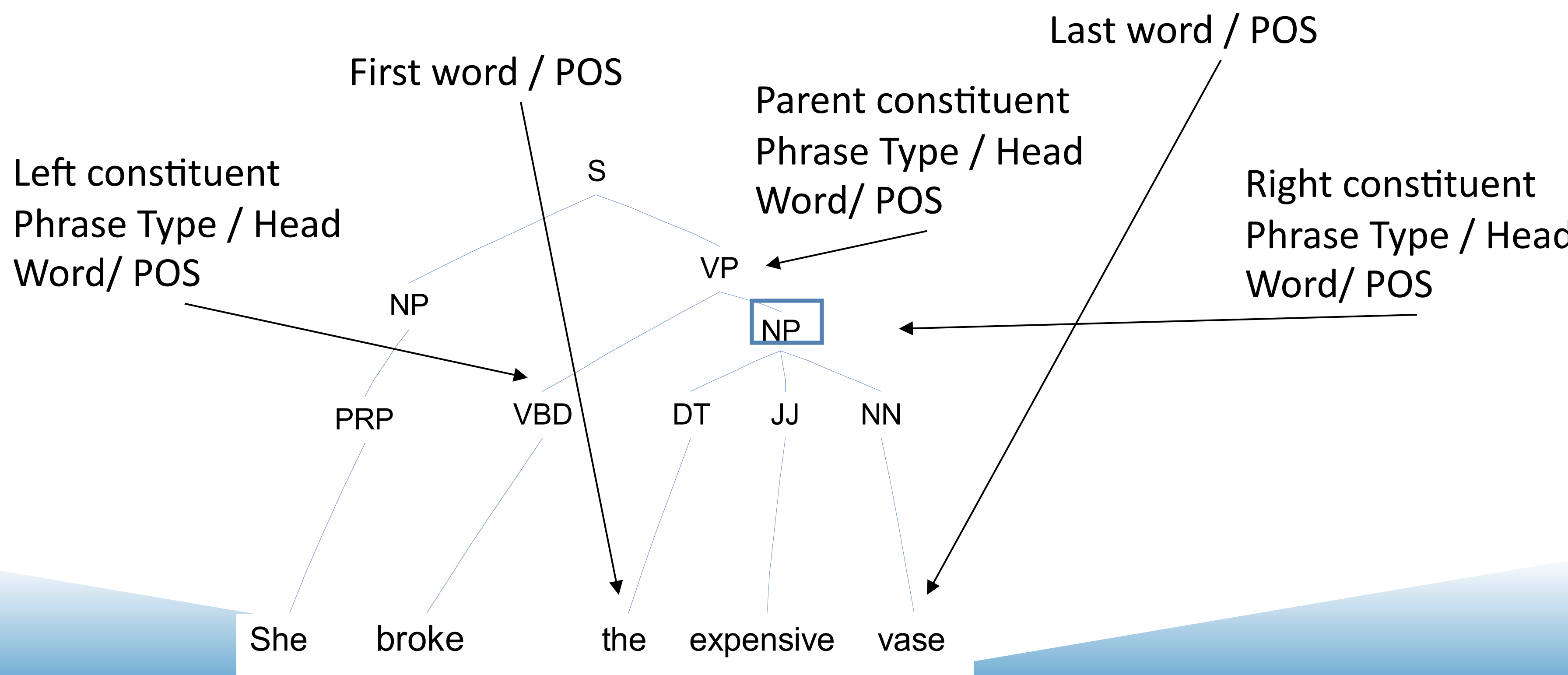


Performance with Baseline Features using the G&J Model

- Better ML: 67.6 → **80.8** using SVMs [Pradhan et al. 04]).
  - Content Word (different from head word)
  - Head Word and Content Word POS tags
  - **NE labels (Organization, Location, etc.)**
  - Structural/lexical context (phrase/words around parse tree)
  - Head of PP Parent
    - If the parent of a constituent is a PP, the identity of the preposition

Pradhan et al. (2004) Features

- More (**31%** error reduction from baseline due to these + Surdeanu et al. features)



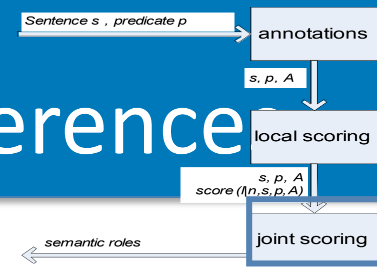
Joint Scoring: Enforcing Hard Constraints

- Constraint 1: Argument phrases do not overlap

  By [$_{A1}$ working [$_{A1}$ hard] , he] ***said*** *, you can achieve a lot.*

  – Pradhan et al. (04) – greedy search for a best set of non-overlapping arguments
  – Toutanova et al. (05) – exact search for the best set of non-overlapping arguments (dynamic programming, linear in the size of the tree)
  – Punyakanok et al. (05) – exact search for best non-overlapping arguments using integer linear programming
- Other constraints ([Punyakanok et al. 04, 05])
  – no repeated core arguments (good heuristic)
  – phrases do not overlap the predicate
  – (*more later*)

Joint Scoring: Integrating Soft Preferences



- Gildea and Jurafsky (02) – a smoothed relative frequency estimate of the probability of frame element multi-sets:

$$P(\{A0, AM_{TMP}, A1, AM_{TMP}\}|hit)$$

  – Gains relative to local model 59.2 → 62.9 FrameNet automatic parses

- Pradhan et al. (04) – a language model on argument label sequences (with the predicate included)

  – Small gains relative to local model for a baseline system 88.0 → 88.9 on core arguments PropBank correct parses

$$P(A0, AM_{TMP}, hit, A1, AM_{TMP})$$

- Toutanova et al. (05) – a joint model based on CRFs with a rich set of joint features of the sequence of labeled arguments (*more later)*
  – Gains relative to local model on PropBank correct parses 88.4 → 91.2 (24% error reduction); gains on automatic parses 78.2 → 80.0

- Also tree CRFs [Cohn & Brunson] have been used

Per Argument Performance

CoNLL-05 Results on WSJ-Test

- Core Arguments (Freq. ~70%)

	Best F_1	Freq.
A0	88.31	25.58%
A1	79.91	35.36%
A2	70.26	8.26%
A3	65.26	1.39%
A4	77.25	1.09%

- Adjuncts (Freq. ~30%)

	Best F_1	Freq.
TMP	78.21	6.86%
ADV	59.73	3.46%
DIS	80.45	2.05%
MNR	59.22	2.67%
LOC	60.99	2.48%
MOD	98.47	3.83%
CAU	64.62	0.50%
NEG	98.91	1.36%

Arguments that need to be improved

Data from Carreras&Màrquez’s slides (CoNLL 2005)