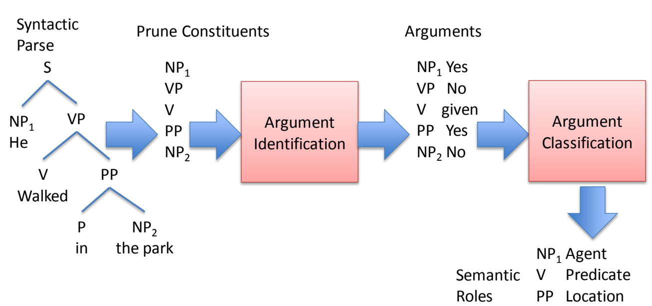
Semantic Role Labeling

Presented to LING-7800
Lee Becker and Shumin Wu

Task

- Given a sentence,
  – Identify predicates and their arguments
  – Automatically label them with semantic roles
- From:
  – Mary slapped John with a frozen trout
- To:
  – [$_{\text{AGENT}}$ Mary] [$_{\text{PREDICATE}}$ slapped] [$_{\text{PATIENT}}$ John] [$_{\text{INSTRUMENT}}$ with a frozen trout]

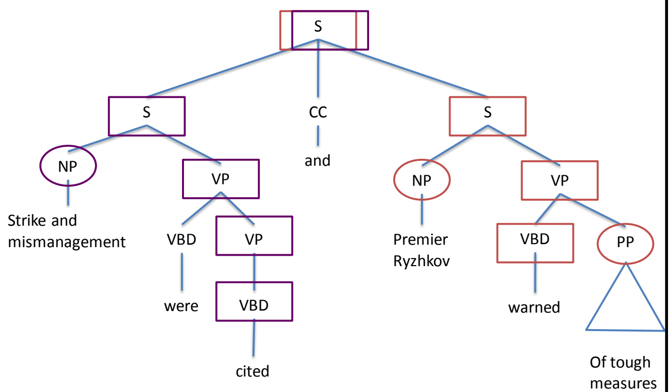
SRL Pipeline



Pruning Algorithm [Xue, Palmer 2004]

- Goal: Reduce the overall number of constituents to label
- Reasoning: Save training time
- Step 1:
  – Designate the predicate as the current node and collect its sisters unless the sister is *coordinated* with the predicate
  – If a sister is a PP also collect its immediate children
- Step 2:
  – Reset current node as the parent node
  – Repeat Steps 1 and 2 until we've reached the top node

Pruning Algorithm [Xue, Palmer 2004]



SRL Training

1. Extract features from sentence, syntactic parse, and other sources for each candidate constituent
2. Train statistical ML classifier to identify arguments
3. Extract features same as or similar to those in step 1
4. Train statistical ML classifier to select appropriate label for arguments
   - Multiclass
   - All vs One

Training Data

- Need Gold Standard:
  - Syntactic Parses (Constituent, Phrase-Based Dependency Based)
  - Semantic Roles (Frame Elements, Arguments, etc)
- Lexical Resources:
  - FrameNet (Baker et al, 1998)
    - 49,000 annotated sentences from the BNC
    - 99,232 annotated frame elements
    - 1462 target words from 67 frames
      - 927 verbs, 339 nouns, 175 adjectives
  - PropBank (Palmer, Kingsbury, Gildea, 2005)
    - Annotation over the Penn Treebank
    - ??? Verb predicates
  - Salsa (Erk, Kowalksi, Pinkal, 2003)
    - Annotation over the German 1.5 million word Tiger corpus
    - FrameNet Semantic roles
  - Various Bakeoffs
    - SemEval
    - CoNLL

Feature Extraction

- The sentence and parses by themselves provide little useful information for selecting semantic role labels
- Need algorithms that derive features from these data that provide some clues about the relationship between the constituent and the sentence as a whole

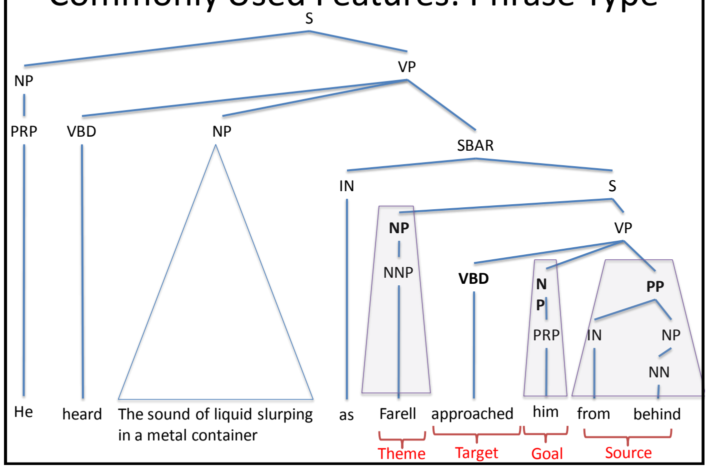
Features: Phrase Type

- Intuition: Different roles tend to be realized by different syntactic categories
- FrameNet Communication_noise frame
  - *Speaker* often is a noun phrase
  - *Topic* typically a noun phrase or prepositional phrase
  - *Medium* usually a prepositional phrase
    - [$_{\text{SPEAKER}}$ The angry customer] yelled at the fast food worker [$_{\text{TOPIC}}$ about his soggy fries] [$_{\text{MEDIUM}}$ over the noisy intercom].

Commonly Used Features: Phrase Type

- Phrase Type indicates the syntactic category of the phrase expressing the semantic roles
- Syntactic categories from the Penn Treebank
- FrameNet distributions:
  - NP (47%) – noun phrase
  - PP (22%) – prepositional phrase
  - ADVP (4%) – adverbial phrase
  - PRT (2%) – particles (e.g. make something *up*)
  - SBAR (2%), S (2%) - clauses

Commonly Used Features: Phrase Type



Commonly Used Features: Governing Category

- Intuition: There is often a link between semantic roles and their syntactic realization as subject or direct object
- *He drove the car over the cliff*
  - Subject NP more likely to fill the agent role
- Grammatical functions may not be directly available in all parser representations

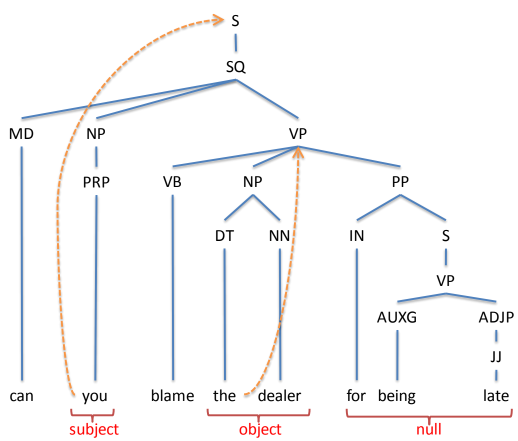
Commonly Used Features: Governing Category

- Approximating Grammatical Function from constituent parse
- Governing Category (aka *gov*)
  - Two values
    - S: subjects
    - VP: object of verbs
  - In practice used only on NPs

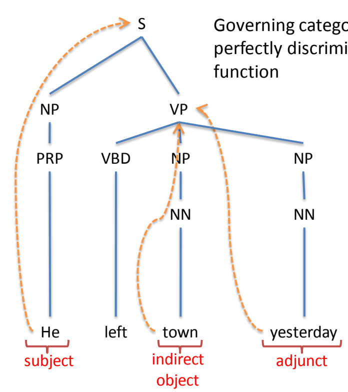
Commonly Used Features: Governing Category

- Algorithm
  - Start with children of NP nodes
  - Traverse links upward until it encounters an S or VP
- NPs under S nodes → subject
- NPs under VP nodes → object

Features: Governing Category

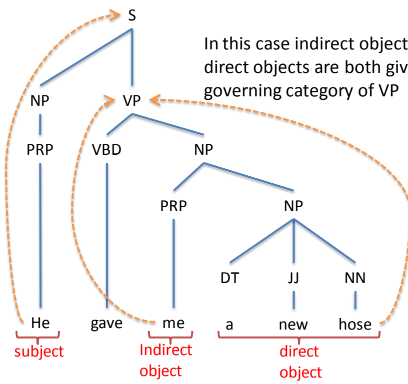


Features: Governing Category



Governing category does not perfectly discriminate grammatical function

Features: Governing Category

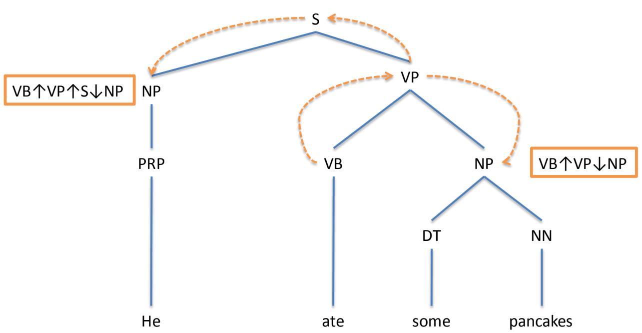


In this case indirect object and direct objects are both given governing category of VP

Features: Parse Tree Path

- Parse Tree Path
  - Intuition: *gov* finds grammatical function independent of target word. Want something that factors in relation to the target word.
  - Feature representation: String of symbols indicating the up and down traversal to go from the target word to the constituent of interest

Features: Parse Tree Path



Features: Parse Tree Path

Frequency	Path	Description
14.2%	VB↑VP↓PP	PP argument/adjunct
11.8	VB↑VP↑S↓NP	subject
10.1	VB↑VP↓NP	object
7.9	VB↑VP↑VP↑S↓NP	subject (embedded VP)
4.1	VB↑VP↓ADVP	adverbial adjunct
3.0	NN↑NP↑NP↓PP	prepositional complement of noun
1.7	VB↑VP↓PRT	adverbial particle
1.6	VB↑VP↑VP↑VP↑S↓NP	subject (embedded VP)
14.2		no matching parse constituent
31.4	Other	none

Features: Parse Tree Path

- Issues:
  - Parser quality (error rate)
  - Data sparseness
    - 2978 possible values excluding frame elements with no matching parse constituent
    - 4086 possible values including total
    - Only 35,138 frame elements identifies as NP, only 4% have path feature without VP or S ancestor [Gildea and Jurafsky, 2002]

Features: Position

- Intuition: grammatical function is highly correlated with position in the sentence
  - Subjects appear before a verb
  - Objects appear after a verb
- Representation:
  - Binary value – does node appear before or after the predicate
- Other motivations [Gildea and Jurafsky, 2002]
  - Overcome errors due to incorrect parses
  - Assess ability to perform SRL without parse trees

Features: Position

Can you blame the dealer for being late?

before after after

Features: Voice

- Intuition: Grammatical function varies with voice
  - Direct objects in active ⇔ Subject in passive
  - He slammed the door.
  - The door was slammed by him.
- Approach:
  - Use passive identifying patterns / templates
    - Passive auxiliary (*to be*, *to get)*
    - Past participle

Features: Head Word

- Intuition: Head words of noun phrases can be be indicative of selectional restrictions on the semantic types of role fillers.
  – Noun Phrases headed by *Bill, brother,* or *he* more likely to be the *Speaker*
  – Those headed by *proposal, story*, or *question* are more likely to be the *Topic*.
- Approach:
  – Most parsers can mark the head word
  – Can employ head words on a constituent parse tree to identify head words

Features: Head Words

- Head Rules – a way of deterministically identifying the head word for a phrase

ADJP	←	NNS QP NN $ ADVP JJ VBN VBG ADJP JJR NP JJS DT FW RBR RBS SBAR RB
ADVP	←	RB RBR RBS FW ADVP TO CD JJR JJ IN NP JJS NN
CONJP	←	CC RB IN
FRAG	→	(NN* | NP) W* SBAR (PP | IN) (ADJP | JJ) ADVP RB
NP, NX	←	(NN* | NX) JJR CD JJ JJS RB QP NP-e NP
PP, WHPP	→	(first non-punctuation after preposition)
PRN	→	(first non-punctuation)
PRT	→	RP
S	←	VP *-PRD S SBAR ADJP UCP NP
VP	→	VBD VBN MD VBZ VB VBG VBP VP *-PRD ADJP NN NNS NP

Sample Head Percolation Rules [Johansson and Nugues]

Features: Subcategorization

- Subcategorization
- Intuition: Knowing the number of arguments to the verb changes the possible set of semantic roles
  – The door closed { subject}
  – The blowing wind closed the door {subject, object}

Features: Argument Set

- Aka: Frame Element Group – set of all roles appearing for a verb in a given sentence
- Intuition: When deciding one role labels it's useful to know their place in the set as a whole
- Representation:
  – {Agent/Patient/Theme}
  – {Speaker/Topic}
- Approach: Not used in training of the system, instead used after all roles are assign to re-rank role assignments for an entire sentence

Features: Argument Order [Fleischman, 2003]

- Description: An integer indicating the position of a constituent in the sequence of arguments
- Intuition: Role labels typically occur in a common order

Can $\underbrace{\text{you}}_{1}$ <u>blame</u> $\underbrace{\text{the dealer}}_{2}$ $\underbrace{\text{for being late?}}_{3}$

- Advantages: independent of parser output, thus robust to parser error

Features: Previous Role [Fleischman, 2003]

- Description: The label assigned by the system to the previous argument.
- Intuition: If we know what's already been labeled we can better know what the current label should be.
- Approach: HMM-style Viterbi search to find best overall sequence

Features: Head Word Part of Speech [Surdeanu et al, 2003]

- Intuition: Penn Treebank POS labels differentiate singular/plural and proper/common nouns. This additional information helps refine the type of noun phrase for a role.

Features: Named entities in Constituents [Pradhan, 2005]

- Intuition: Knowing they type of the entity can allow for better generalization, since unlimited sets of proper names for people, organizations, and locations can make lead to data sparsity.
- Approach: Run a named entity recognizer on the sentences and use the entity label as a feature.
- Representation: Words are identified as a type of entity such as PERSON, ORGANIZATION, LOCATION, PERCENT, MONEY, TIME, and DATE.

Features: Verb Clustering

- Intuition: Semantically similar verbs undergo the same pattern of argument alternation [Levin, 1993]
- Representation: constituent is labeled with a verb class discovered in clustering
  - *He ate the cake*. {verb_class = eat}
  - *He devoured his sandwich*. {verb_class = eat}
- Approach: Perform automatic clustering of verbs based on direct objects
  - ML Approaches:
    - Expectation-Maximization
    - K-means

Features: Head Word of PPs

- Intuition: While prepositions often indicate certain semantic roles (i.e. *in*, *across*, and *toward = location*, *from = source*), prepositions can be used in many different ways.
  - *We saw the play in New York = Location*
  - *We saw the play in February = Time*

Features: First/Last word/POS in constituent

- Intuition: Like with head word of PPs, we want more specific information about an argument than the headword alone.
- Advantages:
  - More robust to parser error
  - Applies to all types of constituents

He was born in the final minutes of 2009

First Word/POS: He / PRN
Last Word/POS: He / PRN

First Word/POS: in/ IN
Last Word/POS: 2009/ CD

Features: Constituent Order

- Intuition:
  - Like argument order, but we want a way to differentiate constituents from non-constituents.
  - Preference should go to constituents closer to the predicate.

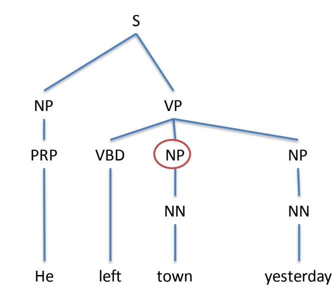
Features: Constituent Tree Distance

- Description: the number of jumps necessary to get from the predicate to the constituent – like a path length
- Intuition: Like the Constituent Order, but factoring in syntactic structure

Features: Constituent Context Features

- Description: Information about the parent and left and right siblings of a constituent
- Intuition: Knowing a constituent's place in the sentence helps determine the role.

Features: Constituent Context Features



Parent Phrase Type	Parent Head Word	Parent Head Word POS	Left Sibling Phrase Type	Left Sibling Head Word	Left Sibling Head Word POS	Right Sibling Phrase Type	Right Sibling Head Word	Right Sibling Head Word POS
VP	left	VBD	None	left	VBD	NP	yesterda y	NN

Features: Temporal Cue Words

- Intuition: Some words indicate time, but are not considered named entities by the named entity tagger.
- Approach:
  - Words are matched in a gloss and included as binary features

Moment	Wink of an eye	…	Around the clock
0	1	…	0

Evaluation

- Precision – percentage of labels output by the system which are correct
- Recall – recall percentage of true labels correctly identified by the system
- F-measure, F_beta – harmonic mean of precision and recall

$$F = \frac{2PR}{P+R}$$

$$F_\beta = \frac{(1+\beta^2)PR}{\beta^2 P + R}$$

Evaluation

- Why all these measures?
  - To keep us honest
  - Together Precision and Recall capture the tradeoffs made in performing a classification task
    - 100% precision is easy on a small subset of the data
    - 100% recall is easy if everything is included
  - Consider a doctor deciding whether or not to perform an appendectomy
    - Can claim 100% precision if surgery is only performed on patients that have been administered a complete battery of tests.
    - Can claim 100% recall if surgery is given to all patients

Evaluation

- Lots of choices when evaluating in SRL:
  - Arguments
    - Entire span
    - Headword only
  - Predicates
    - Given
    - System Identifies

Evaluation

Gold Standard Labels	SRL Output	Full	Head
Arg0: John	Arg0: John	+	+
Rel: mopped	Rel: mopped	+	+
Arg1: the floor	Arg1: the floor	+	+
Arg2: with the dress … Thailand	Arg2: the dress	-	+
Arg0: Mary	Arg0: Mary	+	+
Rel: bought	Rel: bought	+	+
Arg1: the dress	Arg1: the dress	+	+
Arg0: Mary		-	-
rel: studying		-	-
Argm-LOC: in Thailand		-	-
Arg0: Mary	Arg0: Mary	+	+
Rel: traveling	Rel: traveling	+	+
Argm-LOC: in Thailand		-	-

John mopped the floor with the dress Mary bought while studying and traveling in Thailand.

Evaluated on Full Arg Span

Precision
P = 8 correct / 10 labeled = 80.0%

Recall
R = 8 correct / 13 possible = 61.5%

F-Measure
F = P x R = 49.2%

Evaluated on Headword Arg

Precision
P = 9 correct / 10 labeled = 90.0%

Recall
R = 9 correct / 13 possible = 69.2%

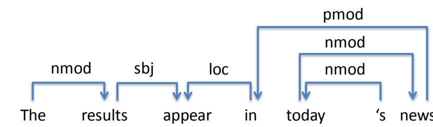
F-Measure
F = P x R = 62.3%

Alternative Representations: Dependency Parse

- Dependency Parses provide much simpler graphs between the arguments
- Much faster computationally

Examples courtesy of Jinho Choi, Retrieving Correct Semantic Boundaries in Dependency Structure

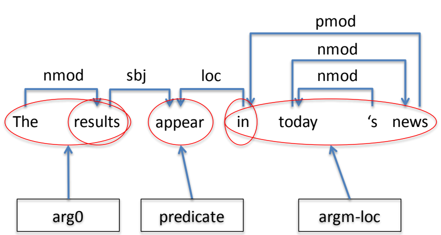
Dependency Parse Example



Examples courtesy of Jinho Choi, Retrieving Correct Semantic Boundaries in Dependency Structure

Argument Identification with Dependency Parse

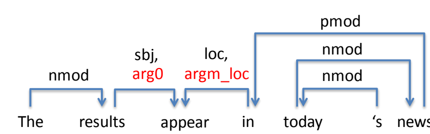
Take the subtree formed by head as arguments:



Examples courtesy of Jinho Choi, Retrieving Correct Semantic Boundaries in Dependency Structure

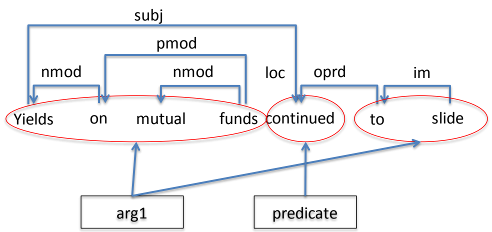
Argument Labeling with Dependency Parse

Integrate dependency label and argument label in training/decoding:



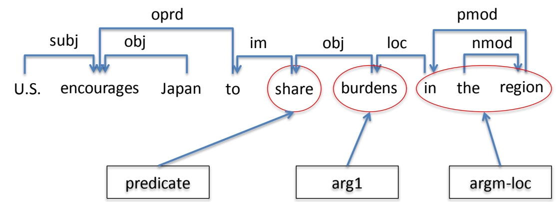
Examples courtesy of Jinho Choi, Retrieving Correct Semantic Boundaries in Dependency Structure

Argument Identification Issues: disjoint argument



Examples courtesy of Jinho Choi, Retrieving Correct Semantic Boundaries in Dependency Structure

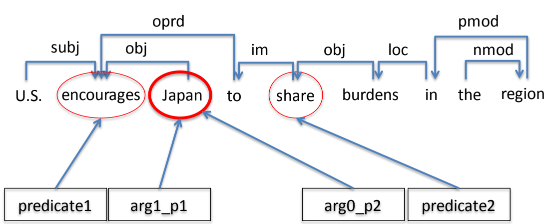
Argument Identification Issues: overlapping arguments



Examples courtesy of Jinho Choi, Retrieving Correct Semantic Boundaries in Dependency Structure

Argument Labeling Issue

With integrated labeling , single phrase cannot serve as argument to 2 different predicate:



Examples courtesy of Jinho Choi, Retrieving Correct Semantic Boundaries in Dependency Structure

Alternative Representations: Syntactic Chunking [Hacioglu et al, 2005]

- Also known as partial parsing
- Classifier trained and used to identify BIO tags
  - B: begin
  - I: inside
  - O: outside

Sales declined 10% to $251.2 million from $278.7 million

Sales	declined	%	to	million	from	million .
B-NP	B-VP	I-NP	B-PP	I-NP	B-PP	I-NP

Alternative Representations: Syntactic Chunking [Hacioglu et al, 2005]

- Features
  - Much overlap
  - Distance
    - distance of the token from the predicate as a number of base phrases
    - same distance as the number of VP chunks
  - Clause Position
    - a binary feature that indicates the token is inside or outside of the clause which contains the predicate

Automatic SRL Issues

- Possible Automatic SRL output:
  - Constituent overlap:

    [$_{\text{ARG0}}$ Mary][$_{\text{PREDICATE}}$ bought][$_{\text{ARG1}}$ the watch[$_{\text{ARGM}}$ from Europe]]
    - Usually not a problem with phrase-structure parse
  - Duplicate argument type (rarely legal in Propbank):

    [$_{\text{ARG0}}$ Mary][$_{\text{PREDICATE}}$ bought][$_{\text{ARG1}}$ John][$_{\text{ARG1}}$ the watch]
    - Infrequent when overall SRL accuracy is high

Linear Programming Problem

- Typical form:

Maximize:

$$c_1x_1 + c_2x_2 + \ldots + c_nx_n$$

Given:

$$a_{11}x_1 + a_{12}x_2 + \ldots a_{1n}x_n \le b_1$$
$$a_{22}x_2 + \ldots a_{2n}x_n \le b_1$$
$$a_{m1}x_1 + a_{m2}x_2 + \ldots a_{mn}x_n \le b_m$$
$$x_i \ge 0$$

- Integer Linear Programming:

$$x_1 \in (0, 1, 2\ldots)$$

Constraint output with Integer Linear Programming

- For an argument candidate (S^i), suppose the classifier returns a score for each argument type (c^i), we can try to optimize the score sum over all argument candidates:

$$\arg\max \sum_{i=1}^{M} score(S^i = c^i)$$

- Given each token in a sentence can only belong to at most 1 argument:

$$\sum_{i=1}^{M} t_j \in S_i \le 1$$

- Each argument type can only occur at most once for a predicate:

$$\sum_{i=1}^{M} (A_j = A_{S_i}) \le 1$$

Punyakanok, et al, Semantic Role Labeling via Integer Linear Programming Inference