The Task of
Semantic Role Labeling

Automatic SRL

- Define an algorithm that will process text and recognize roles for each verb
- Assume previous levels of Natural Language Processing (NLP) on text
  - Part-of-speech (POS) tagging,
  - Parse trees, dependency trees
- Machine Learning classification approaches are typical

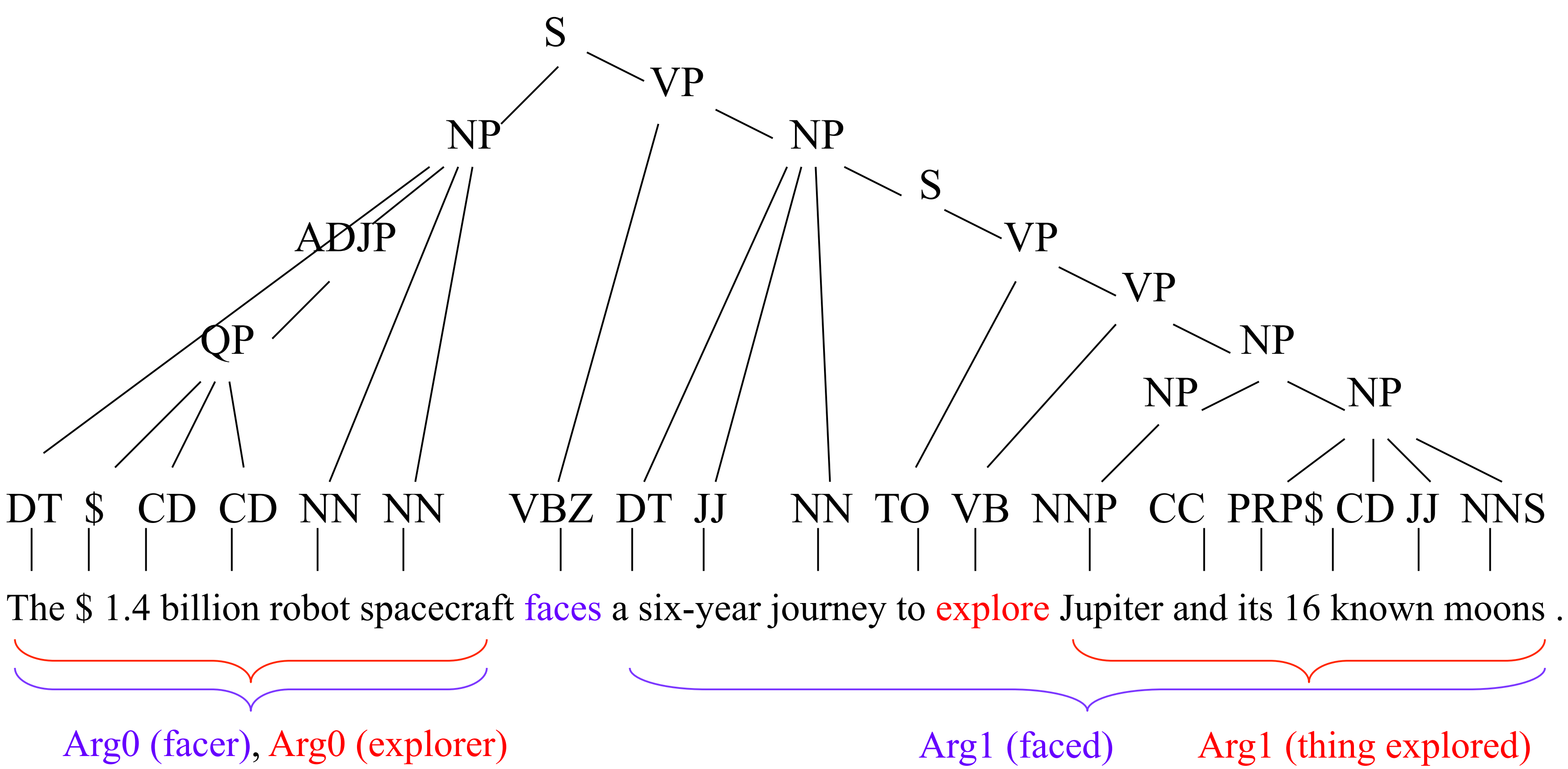
Machine Learning Approach

- Given a verb in a sentence, the problem is to find and label all arguments
- Reformulate as a classification task: For each constituent in the parse tree of the sentence, label it as to what argument, if any, it is for the verb

- For each constituent, define features of semantic roles
  - Each feature describes some aspect of a text phrase that can help determine its semantic role of a verb
    - Examples include what the verb is, POS tags, position in parse tree, etc.
- Machine Learning process:
  - Training:
    - Use annotated corpus of semantic roles with features and semantic role label
      - PropBank or FrameNet
    - ML training program uses examples to produce decision algorithm
  - Classification:
    - Run decision algorithm on text phrases and it will decide which, if any, semantic role it plays with respect to a verb

Parse Tree Constituents

- Each syntactic constituent is a candidate for labeling
- Define features from sentence processed into parse tree with Part-of-Speech tags on words

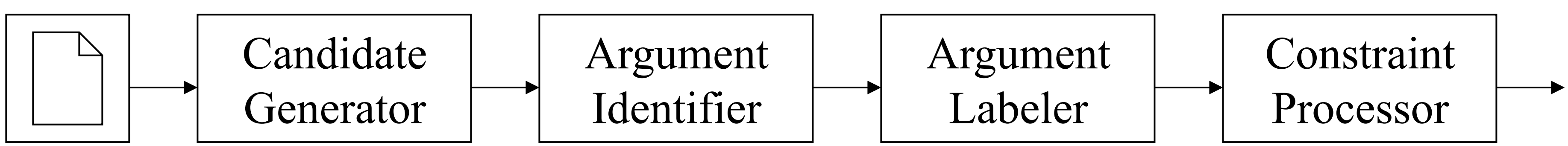


Difficulties for classification

- For each verb in a sentence, the number of constituents in the parse tree are large compared to the number of semantic roles
  - Can be hundreds of constituents eligible to be labeled a role
  - Leads to the problem of too many "negative" examples
- What should the features be?
  - Words are typically the features for an NLP problem
  - Need more about the syntactic structure as well as other potential clues
  - Typical number of features can be up to 20,000, requiring a classification algorithm that is robust for large numbers of features

Typical architecture with 2 step classifier

- Steps of the architecture
  - Candidate Generator: filter out implausible constituents from the parse trees
  - Argument Identifier: use a machine learning classifier to decide if each of the remaining constituents is an argument to the verb
    - Tuned to solve the too many negative example problem
  - Argument Labeler: N binary classifiers, each producing a probability estimate of whether an argument should have that label (Arg0-Arg5, ArgM's, etc.)
  - Do some final constraint processing



Typical Argument Features

- These features are defined for each constituent:
- PREDICATE: The predicate word from the training data.
  - “face” and “explore”
  - Usually stemmed or lemmatized
- PHRASE TYPE: The phrase label of the argument candidate.
  - Examples are NP, S, for phrases, or may be POS tag if a single word
- POSITION: Whether the argument candidate is before or after the predicate.
- VOICE: Whether the predicate is in active or passive voice.
  - Passive voice is recognized if a past participle verb is preceded by a form of the verb “be” within 3 words.
- SUBCATEGORY: The phrase labels of the children of the predicate’s parent in the syntax tree.
  - subcat of “faces” is “VP -> VBZ NP”

Argument Features

- PATH: The syntactic path through the parse tree from the argument constituent to the predicate.
  - Arg0 for “faces”: NP -> S -> VP -> VBZ
- HEAD WORD: The head word of the argument constituent
  - Main noun of NP (noun phrase)
  - Main preposition of PP (prepositional phrase)
- Many additional features
  - Head Word POS: The part of speech tag of the head word of the argument constituent.
  - Temporal Cue Words: Special words occurring in ArgM-TMP phrases.
  - Governing Category: The phrase label of the parent of the argument.
  - Grammatical Rule: The generalization of the subcategorization feature to show the phrase labels of the children of the node that is the lowest parent of all arguments of the predicate.

SRL problem constraints

- Results of the labeling classifier are probabilities for each label that is labels that constituent
- Use these with constraints to assign a label
  - Two constituents cannot have the same argument label,
  - A constituent cannot have more than one label
  - If two constituents have (different) labels, they cannot have any overlap,
  - No argument can overlap the predicate.

CoNLL-2005 Shared Task

- Each year, CoNLL (Conference on Natural Language Learning) defines a task to develop some aspect of natural language processing with systems that use machine learning.
  - Provides data for training and developing systems for about 3 months
  - Then provides test data; everyone runs their system and returns the results for scoring
  - Competitive in that scores are published in a comparative way
  - Collaborative in that a session of the annual conference is devoted to discussion of the progress in this task
    - Novel approached are encouraged
- The 2005 shared task evaluated machine learning SRL systems based on full parse information
  - Best results:

	Precision	Recall	$F_{\beta=1}$
Koomen et al	80.05%	74.83%	77.35

Current Direction of SRL

- Best English SRL results combining parse trees or combining the parsing task with the SRL task (joint inference) are at just F-measure of 80 - 82
- CoNLL 2009 shared task is SRL again, but systems combined dependency parsing with semantic role labeling.
  - Joint detection of syntactic and semantic dependencies
  - Richer syntactic dependency set to aid in semantic processing
  - English, Catalan, Chinese, Czech, German, Japanese, Spanish
- Question: Can applications make good use of SRL?
  - SRL tools are not as generally available as good parsing systems
  - Results are not as accurate as POS tagging (~97) or parsing (~92)
  - But there are systems requiring the semantics in general domain text that have used SRL to give semantic representations
    - IBM's Watson Question Answering system