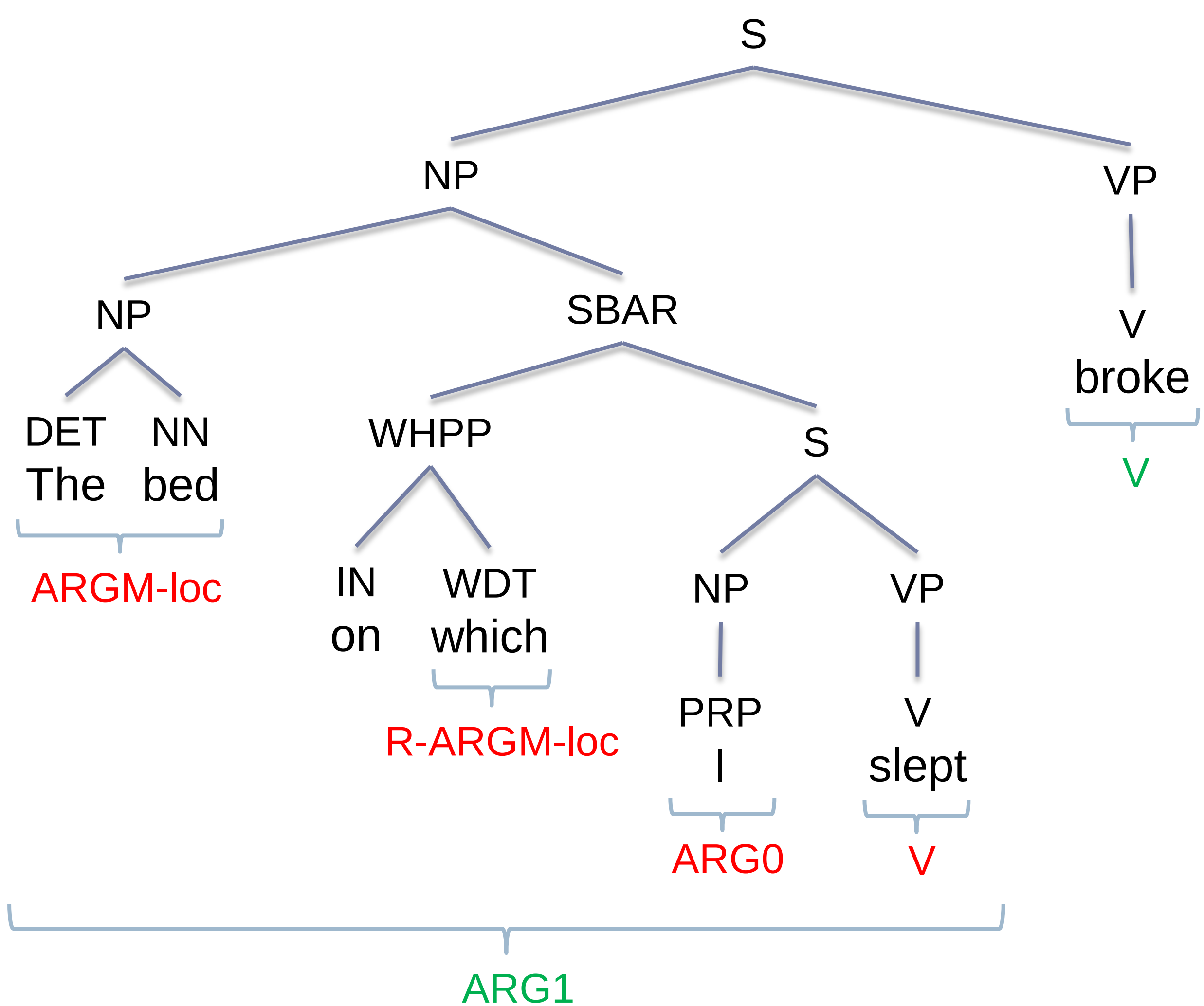
Semantic Role Labeling Tutorial: Part 2

Supervised Machine Learning methods

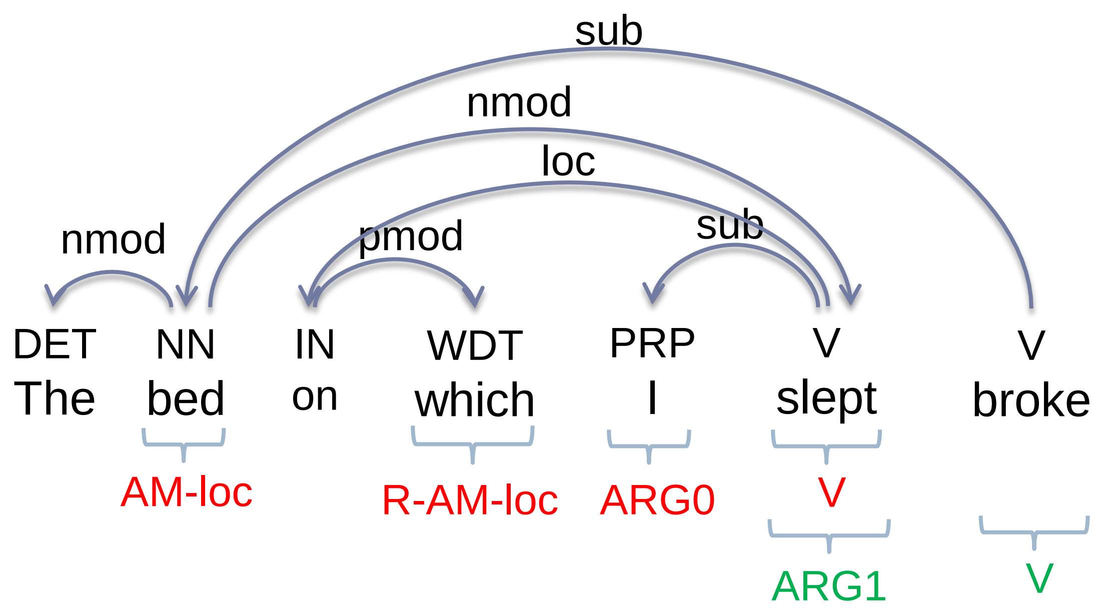
Shumin Wu

Computational Language and EducAtion Research
CLEAR

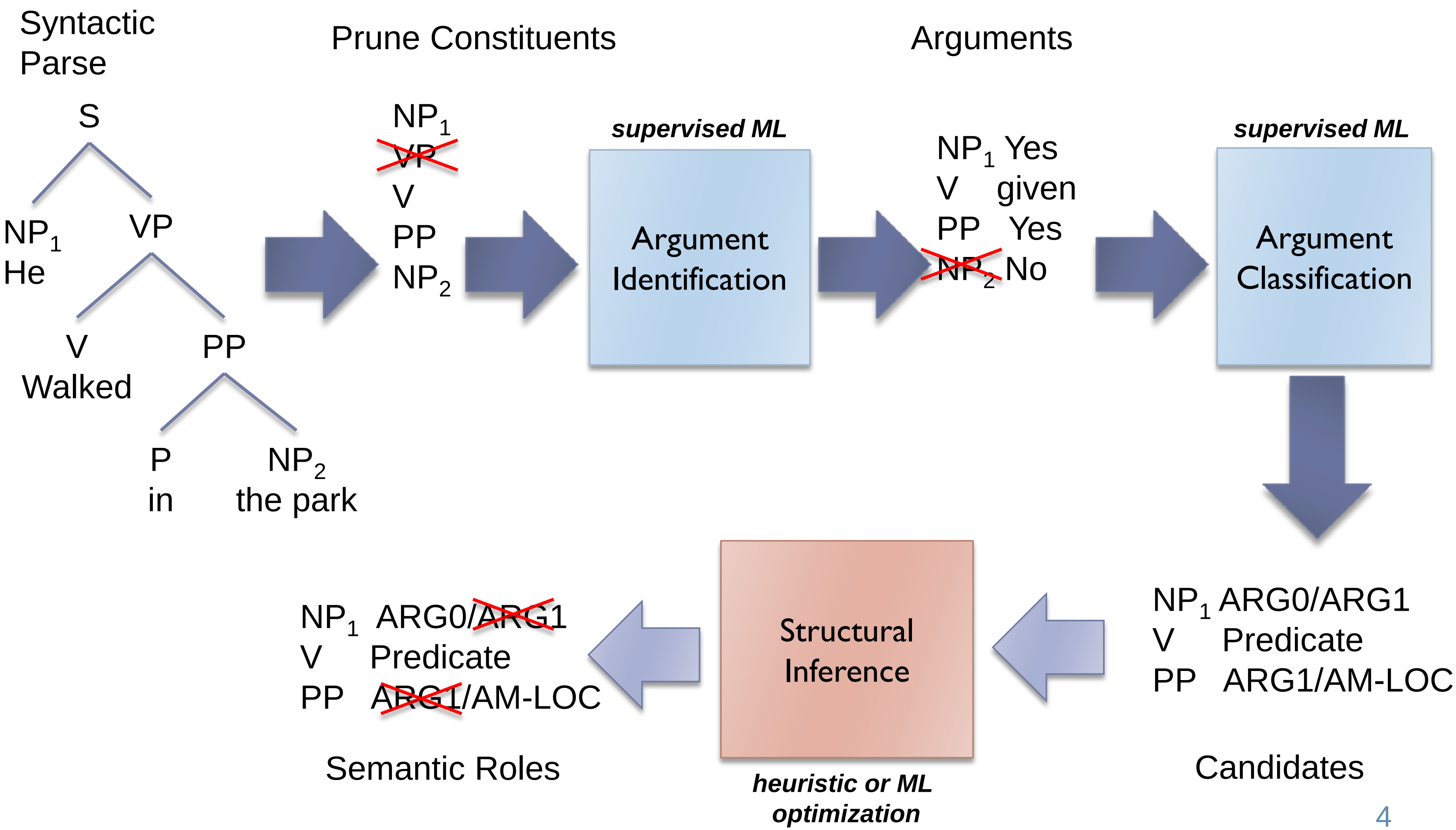
SRL on Constituent Parse



SRL on Dependency Parse

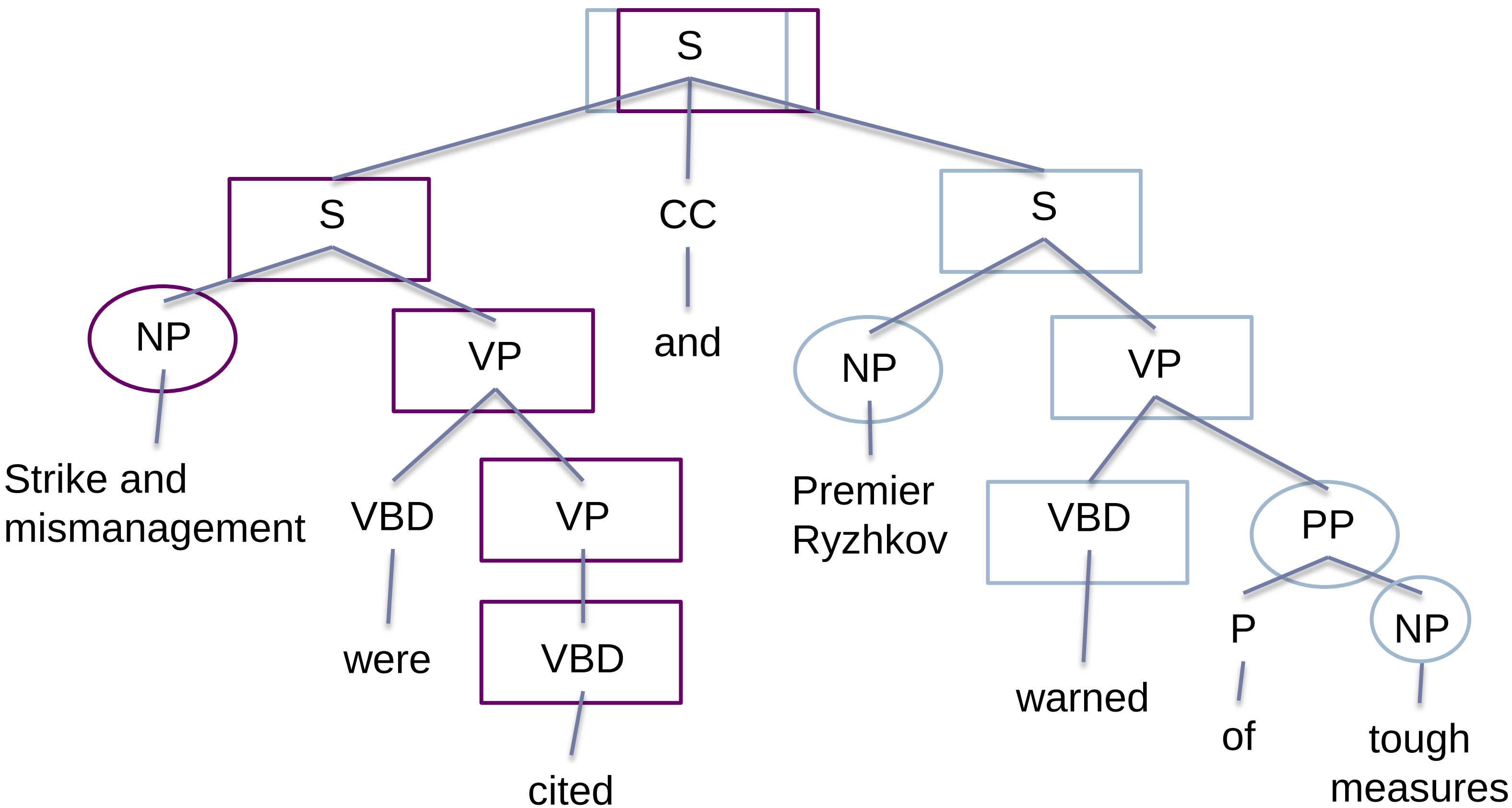


SRL Supervised ML Pipeline



Pruning Algorithm [Xue, Palmer 2004]

- For the predicate and each of its ancestors, collect their sisters unless the sister is *coordinated* with the predicate
- If a sister is a PP also collect its immediate children



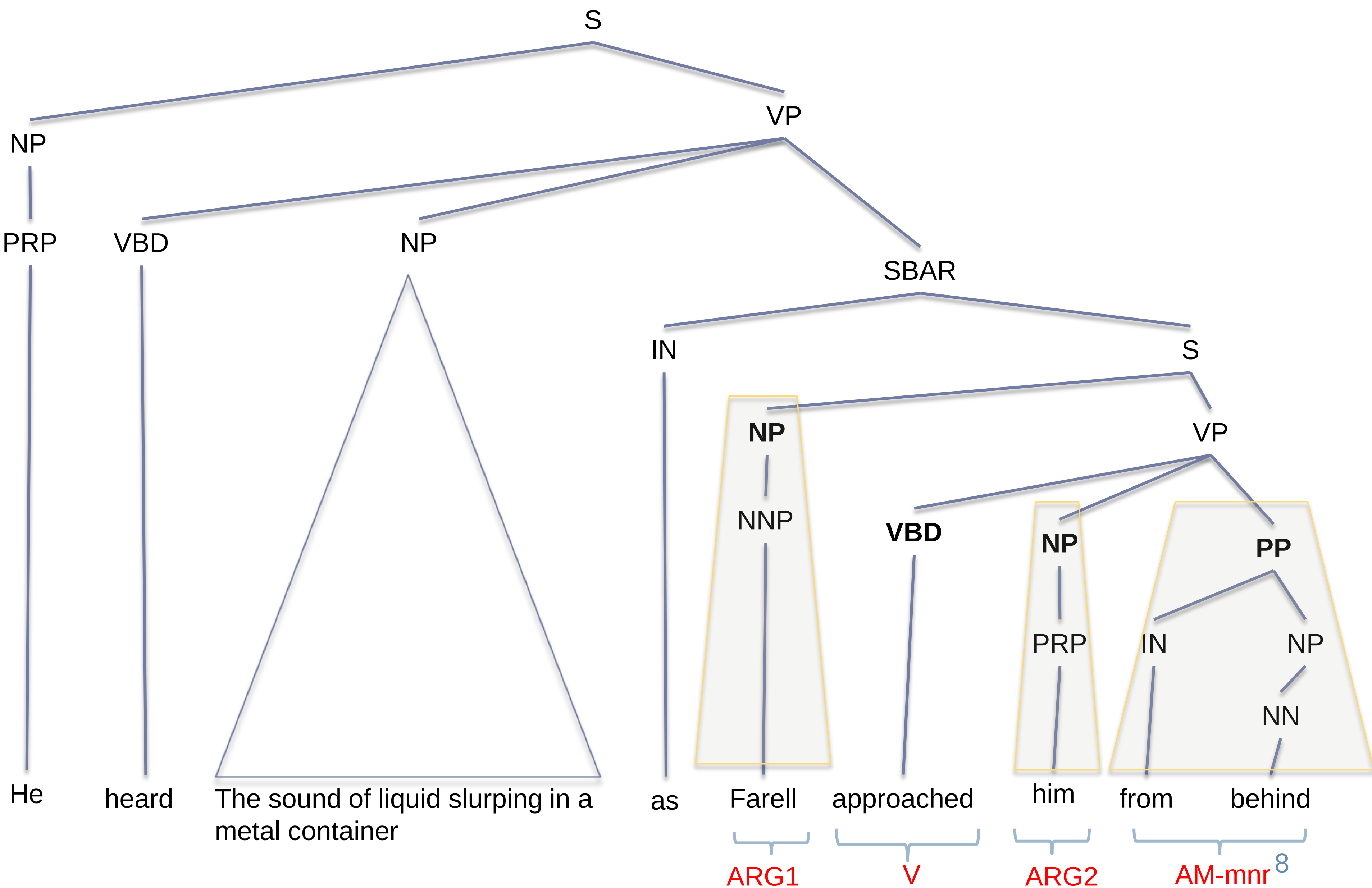
ML for Argument Identification/Labeling

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
   - SVM, Linear (MaxEnt, LibLinear, etc), structured (CRF) classifiers
   - All vs one, pairwise, structured multi-label classification

Commonly Used Features: Phrase Type

- Intuition: different roles tend to be realized by different syntactic categories
- For dependency parse, the dependency label can serve similar function
- 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

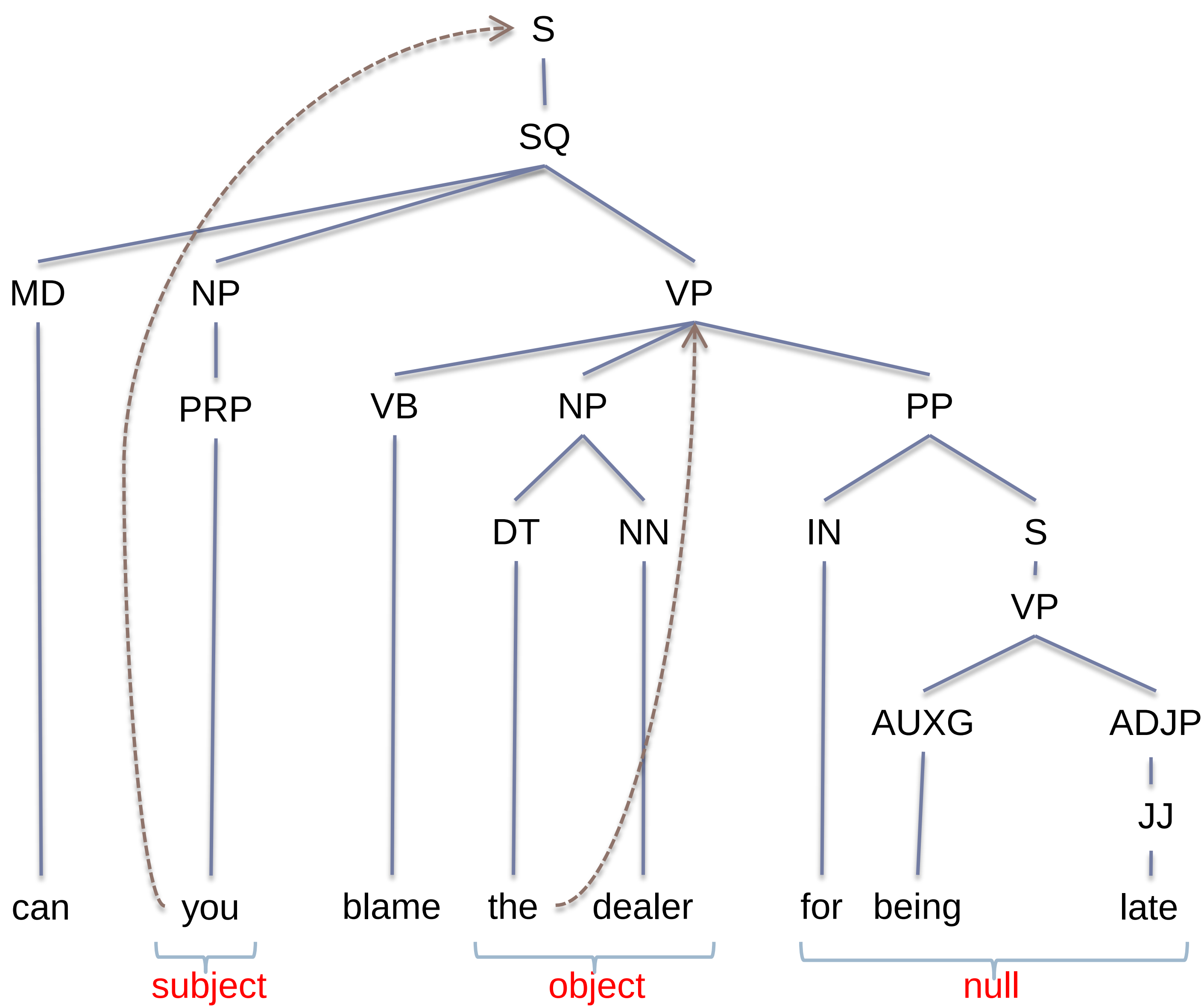
Commonly Used Features: Phrase Type



Features: Governing Category

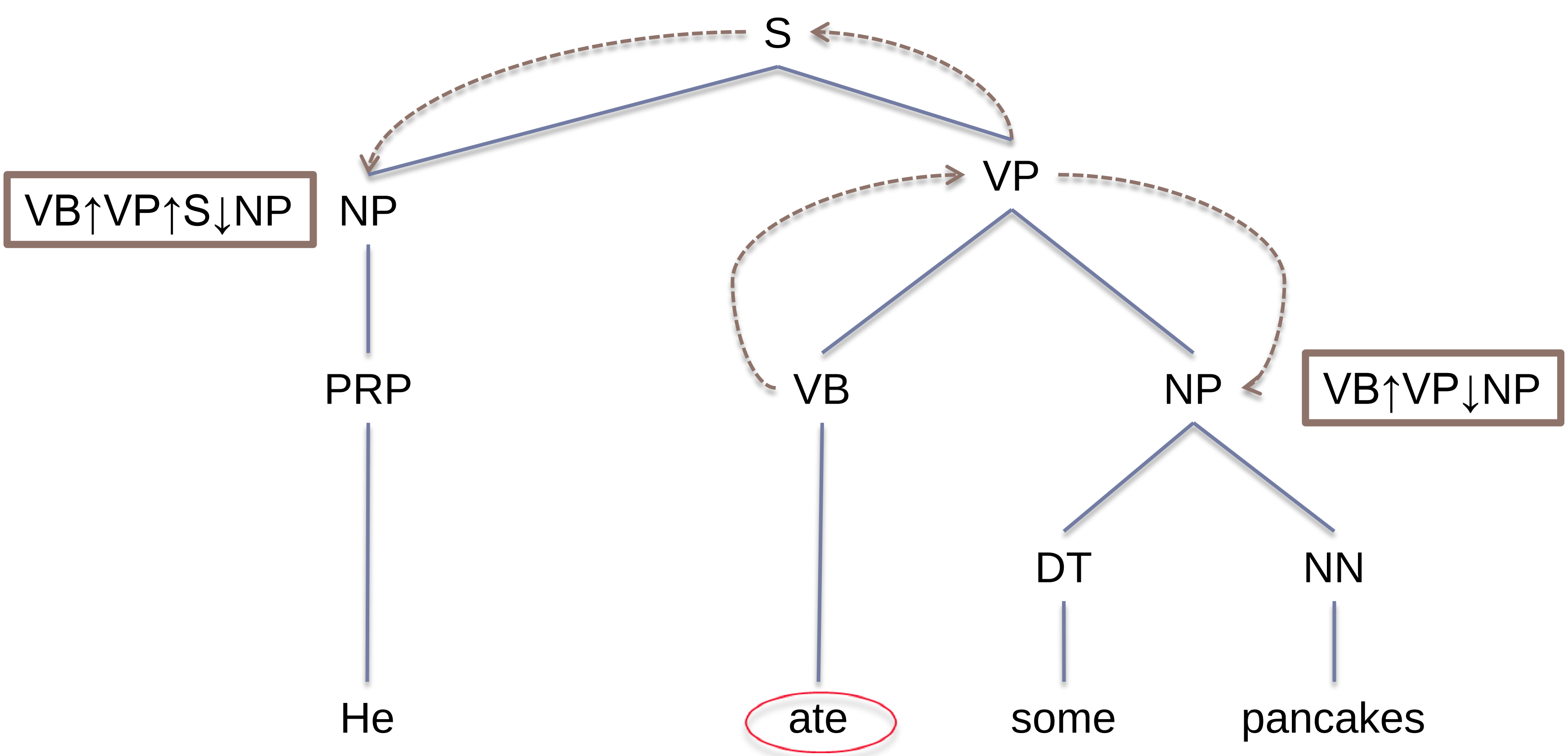
- Intuition: There is often a link between semantic roles and their syntactic realization as subject or direct object
- *He drove the car over the cliff*
  - Subject NP more likely to fill the agent role
- Approximating grammatical function from parse
  - Function tags in constituent parses (typically not recovered in automatic parses)
  - Dependency labels in dependency parses

Features: Governing Category



Features: Parse Tree Path

- Intuition: need a feature 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
- For dependency parses, use dependency path



Features: Parse Tree Path

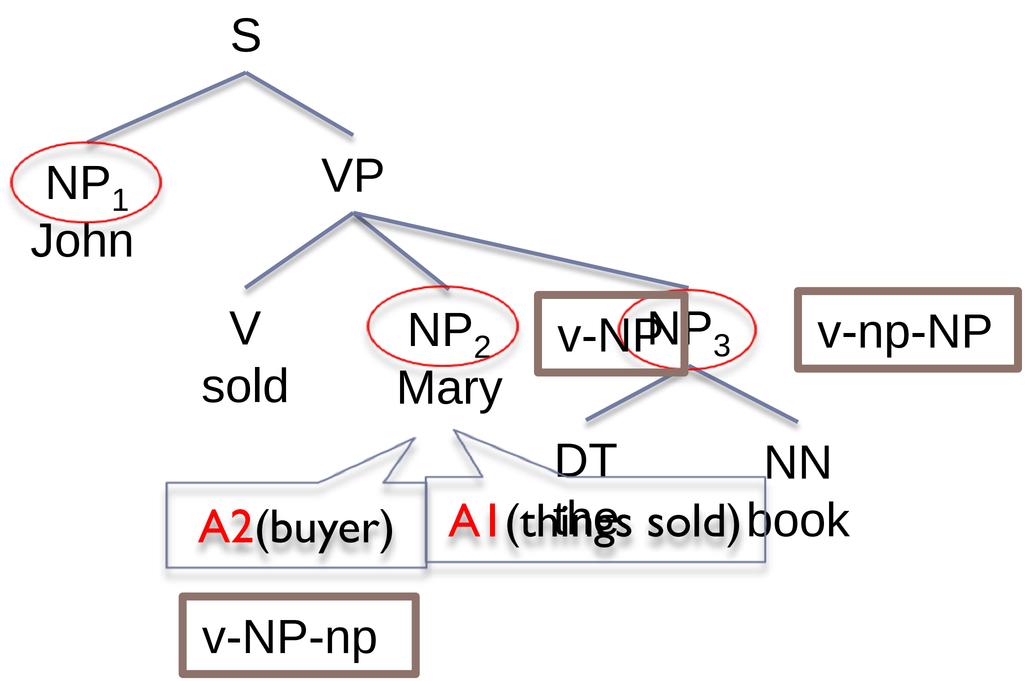
Frequency	Path	Description
14.2%	VB↑VP↓PP	PP argument/adjunct
11.8	VB↑VP↑S↓NP	subject
10.1	VB↑VP↓NP	object
7.9	VB↑VP↑VP↑S↓NP	subject (embedded VP)
4.1	VB↑VP↓ADVP	adverbial adjunct
3.0	NN↑NP↑NP↓PP	prepositional complement of noun
1.7	VB↑VP↓PRT	adverbial particle
1.6	VB↑VP↑VP↑VP↑S↓NP	subject (embedded 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
      - Compress path by removing consecutive phrases of the same type, retain only clauses in path, etc
    - 4086 possible values including total of 35,138 frame elements identifies as NP, only 4% have path feature without VP or S ancestor [Gildea and Jurafsky, 2002]

Features: Subcategorization

- List of child phrase types of the VP
  - highlight the constituent in consideration
- Intuition: Knowing the number of arguments to the verb constrains the possible set of semantic roles
- For dependency parse, collect dependents of predicate



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

$$\text{Can } \underbrace{\text{you}}_{\text{before}} \ \underline{\text{blame}} \ \underbrace{\text{the dealer}}_{\text{after}} \ \underbrace{\text{for being late?}}_{\text{after}}$$

Features: Voice

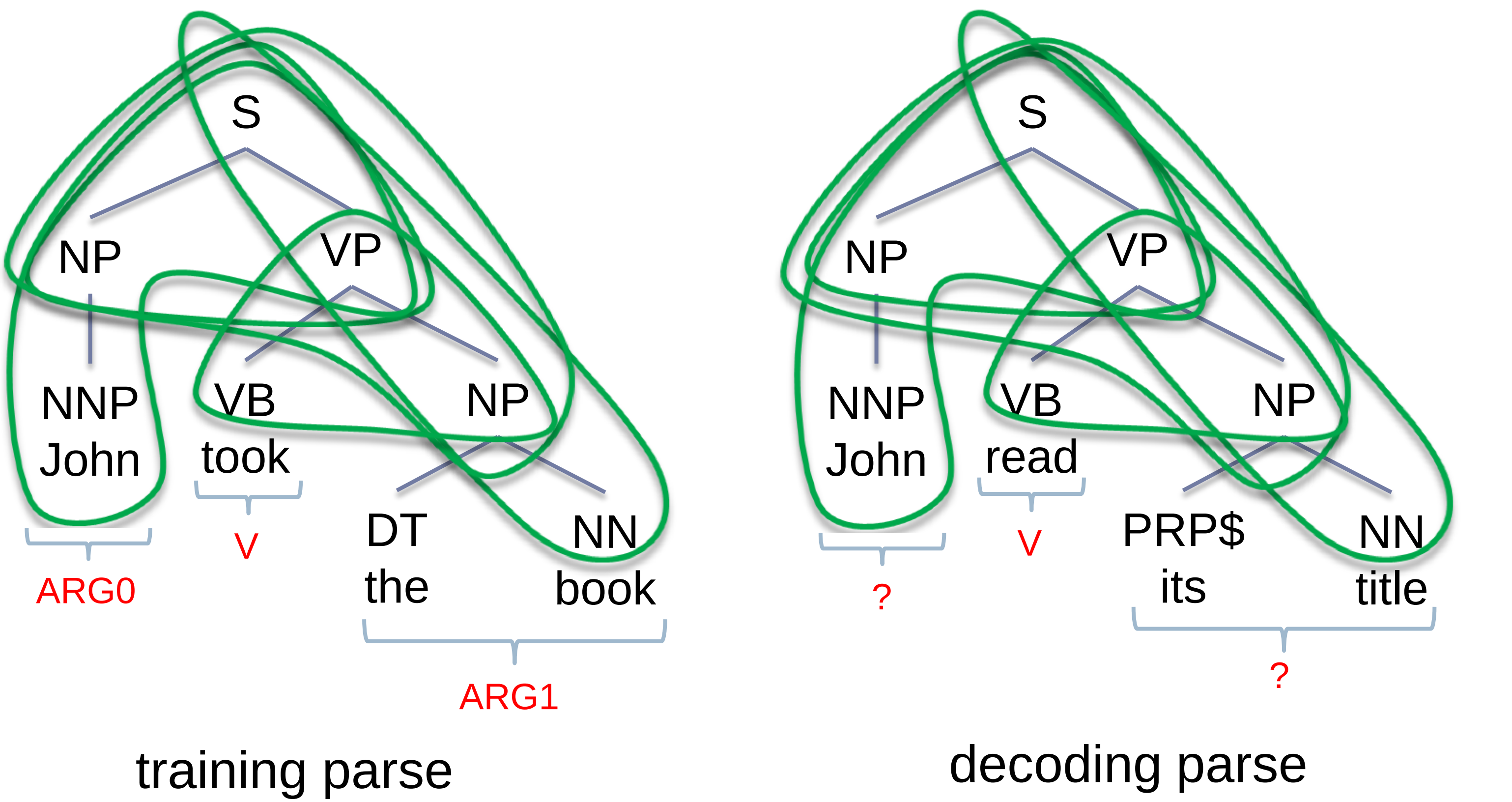
- Intuition: Grammatical function varies with voice
  - Direct objects in active ⇔ Subject in passive

    He slammed **the door**.

    The door was slammed by him.
- Approach:
  - Use passive identifying patterns / templates (language dependent)
    - Passive auxiliary (*to be, to get), past* participle
    - *bei* construction in Chinese

Features: Tree kernel

- Compute sub-trees and partial-trees similarities between training parses and decoding parse



Features: Tree kernel

- Does not require exact feature match
  - Advantage when training data is small (less likely to have exact feature match)
- Well suited for kernel space classifiers (SVM)
  - All possible sub-trees and partial trees do not have to be enumerated as individual features
  - Tree comparison can be made in polynomial time even when the number of possible sub/partial trees are exponential

More Features

- Head word
  - Head of constituent
- Name entities
- Verb cluster
  - Similar verbs share similar argument sets
- First/last word of constituent
- Constituent order/distance
  - Whether certain phrase types appear before the argument
- Argument set
  - Possible arguments in frame file
- Previous role
  - Last found argument type
- Argument order
  - Order of arguments from left to right

Nominal Predicates

- Verb predicate annotation doesn't always capture fine semantic details:

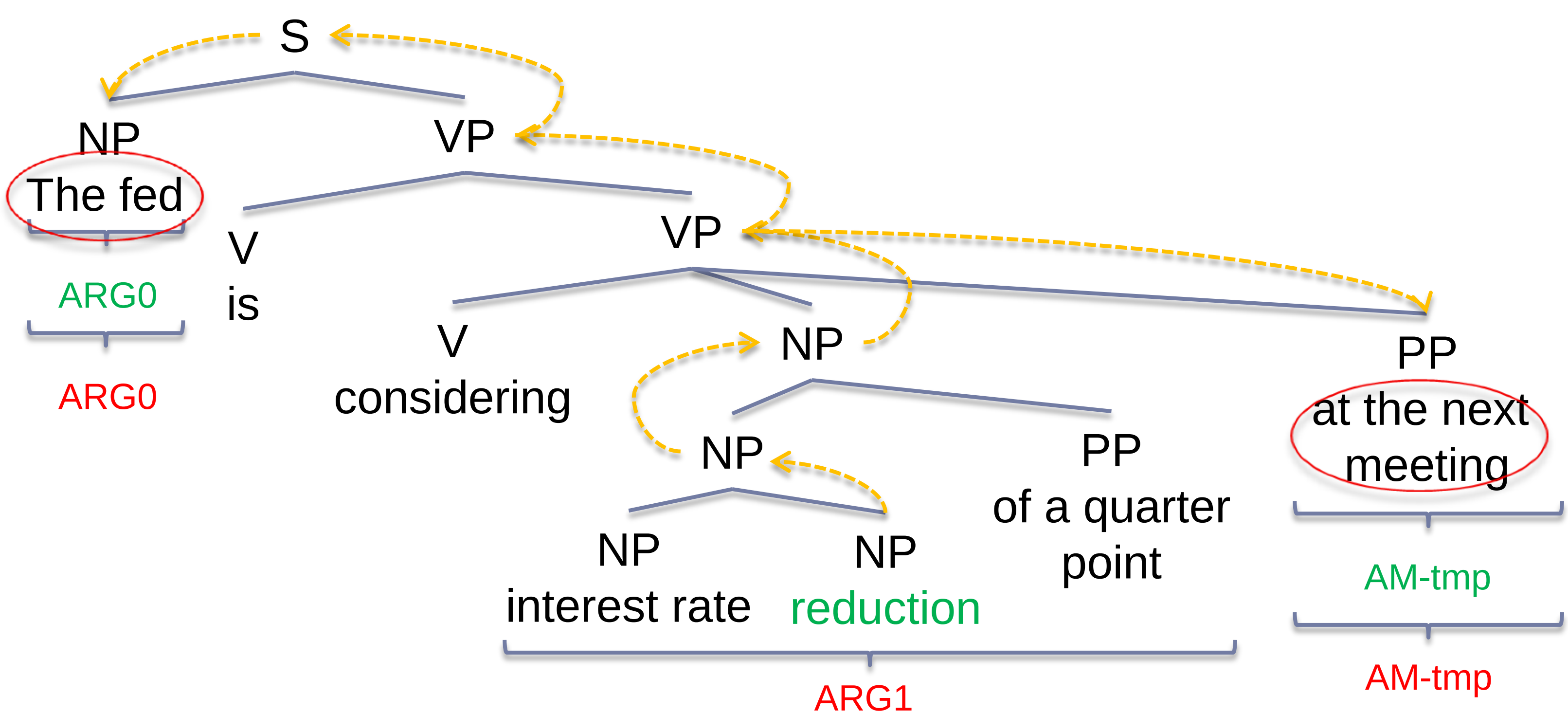
ARG0 V ARG1 AM-tmp

The fed is *considering* interest rate *reduction* of a quarter point at the next meeting

ARG0 ARG1 V-nom ARG2-ext AM-tmp

Arguments of Nominal Predicates

- Can be harder to classify because arguments are not as well constrained by syntax



- Find the "supporting" verb predicate and its argument candidates
  - Usually under the VP headed by the verb predicate and is part of an argument to the verb

Structural Inference

- Take advantage of predicate-argument structures to re-rank argument label set
  - Arguments should not overlap

    Can you blame the dealer for being late?

    ARG0

    ARG1: 0.5

    ARG2: 0.8

    ARG1: 0.6

  - Numbered arguments (arg0-5) should not repeat

    John sold Mary the book

    ARG0

    ARG1: 0.6
    ARG2: 0.4

    ARG1: 0.8
    ARG2: 0.2

  - R-arg[type] and C-arg[type] should have an associated arg[type]

    The bed on which I slept broke

    not arg: 0.6
    AM-loc: 0.4

    R-AM-loc

    ARG0

Structural Inference Methods

- Optimize log probability of label set $(\sum_{i=1}^{n} \log\left(p(A_i)\right) / n)$
  - Beam search
  - Formulate into integer linear programming (ILP) problem
- Re-rank top label sets that conform to constraints
  - Choose n-best label sets
  - Train structural classifier (CRF, etc)

SRL ML Notes

- Syntactic parse input
  - Training parse accuracy needs to match decoding parse accuracy
    - Generate parses via cross-validation
  - Cross-validation folds needs to be selected with low correlation
    - Training data from the same document source needs to be in the same fold
- Separate stages of constituent pruning, argument identification and argument labeling
  - Constituent pruning and argument identification reduce training/decoding complexity, but usually incurs a slight accuracy penalty

Linear Classifier Notes

- Popular choices: LibLinear, MaxEnt, RRM
- Perceptron model in feature space
  - each *feature*$_j$ contributes positively or negatively to a *label*$_i$

$$L_i = sign(w_{i,0} + \sum_j f_j w_{i,j})$$

- How about position and voice features for classifying the agent?

  He *slammed the door.*

  The door was slammed by ***him****.*

  - Position (*left*): positive indicator since active construction is more frequent
  - Voice (*active*): weak positive indicator by itself (agent can be omitted in passive construction)
- Combine the 2 features as a single feature
  - *left-active* and *right-passive* are strong positive indicators
  - *left-passive* and *right-active* are strong negative indicators

Support Vector Machine Notes

- Popular choices: LibSVM, SVM^{light}
- Kernel space classification (linear kernel example)
  - The correlation (c_j) of the features of the input sample with each training $sample_j$ contributes positively or negatively to a $label_i$

$$L_i = sign(w_{i,0} + \sum_j c_j w_{i,j})$$

- Creates $n \times n$ dense correlation matrix during training (n is the size of training samples)
  - Requires a lot of memory during training for large corpus
    - Use a linear classifier for argument identification
    - Train base model with a small subset of samples, iteratively add a portion of incorrectly classified training samples and retrain
  - Decoding speed not as adversely affected
    - Trained model typically only has a small number of "support vectors"
- Tend to perform better when training data is limited

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

- Lots of choices when evaluating in SRL:
  - Arguments
    - Full span (CoNLL-2005)
    - Headword only (CoNLL-2008)
  - Predicates
    - Given (CoNLL-2005)
    - System Identifies (CoNLL-2008)
    - Verb and nominal predicates (CoNLL-2008)

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: with 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 **Head** word 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**%

Applications

- Question & answer systems

Who did *what* to *whom* at *where?*

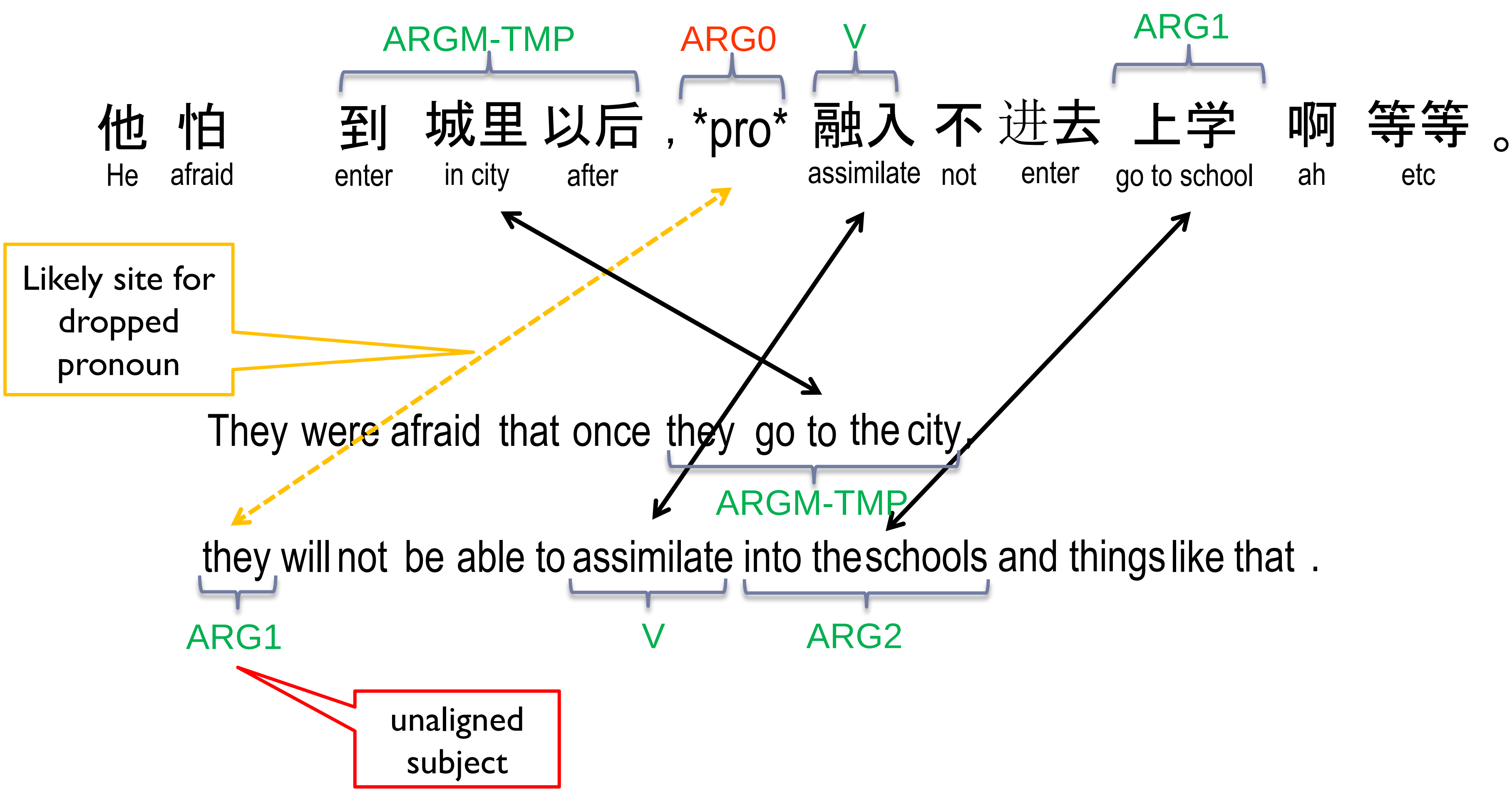
$$\underbrace{\text{The police officer}}_{\text{ARG0}}\ \underbrace{\text{detained}}_{\text{V}}\ \underbrace{\text{the suspect}}_{\text{ARG2}}\ \underbrace{\text{at the scene of the crime}}_{\text{AM-loc}}$$

Multilingual Applications

- Machine translation generation/evaluation

Multilingual Applications

- Identifying/recovering implicit arguments across language
  - Chinese dropped pronoun



SRL Training Data, Parsers

Training Data (Treebank and PropBank):

- LDC

  http://www.ldc.upenn.edu/

Parsers:

- Collins Parser

  http://people.csail.mit.edu/mcollins/code.html

- Charniak Parser

  http://cs.brown.edu/people/ec/#software

- Berkeley Parser

  http://code.google.com/p/berkeleyparser/

- Stanford Parser (includes dependency conversion tools)

  http://nlp.stanford.edu/downloads/lex-parser.shtml

- ClearNLP (dependency parser and labeler, Apache license)

  https://code.google.com/p/clearnlp/

Some SRL systems on the Web

Constituent Based SRL:

- ASSERT
  - one of the top CoNLL-2005 system, extended to C-ASSERT for Chinese SRL)

  http://cemantix.org/software/assert.html
- Senna (GPL license)
  - fast implementation in C

  http://ml.nec-labs.com/senna/
- SwiRL
  - one of the top CoNLL-2005 system

  http://www.surdeanu.info/mihai/swirl/
- UIUC SRL Demo
  - based on the top CoNLL-2005 system w/ ILP argument set inference

  http://cogcomp.cs.illinois.edu/demo/srl/

Dependency Based SRL:

- ClearNLP (dependency parser and labeler, Apache license)
  - state-of-the-art dependency based SRL (comparable to top CoNLL-2008 system)
  - models for OntoNotes and medical data, actively maintained

  https://code.google.com/p/clearnlp/

References

A. Berger , S. Della Pietra and V. Della Pietra, *A Maximum Entropy approach to Natural Language Processing*. Computational Linguistics, 1996

X. Carreras and Lluis Marquez, *Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling*. http://www.lsi.upc.edu/~srlconll/st05/st05.html, 2005

C.-C. Chang and C.-J. Lin. *LIBSVM : a library for support vector machines*. ACM Transactions on Intelligent Systems and Technology, 2011

J. D. Choi, M. Palmer, and Ni Xue. *Using parallel propbanks to enhance word-alignments*. ACL-LAW, 2009

J. D. Choi , *Optimization of Natural Language Processing Components for Robustness and Scalability*, Ph.D. Thesis, CU Boulder, 2012.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. *Liblinear: A library for large linear classification*. Journal of Machine Learning Research, 2008

P. Fung, Z. Wu, Y. Yang, and D.Wu. *Automatic learning of chinese-english semantic structure mapping*. ACL-SLT, 2006.

P. Fung, Z. Wu, Y. Yang, and D.Wu. *Learning bilingual semantic frames: Shallow semantic parsing vs. semantic role projection*. TMI, 2007

D Gildea and D. Jurafsky. *Automatic labeling of semantic roles*. Computational Linguistics, 2002

R. Johansson and P. Nugues. *Extended Constituent-to-dependency Conversion for English*. NODALIDA 2007, 2007.

C. Lo and D.Wu. *Meant: An inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility via semantic frames*. ACL-HLT, 2011

A. Moschitti, D. Pighin, and R. Basili. *Tree kernels for semantic role labeling*. Computational Linguistics, 2008

V. Punyakanok, D. Roth and Wen-tau Yih. *The Importance of Syntactic Parsing and Inference in Semantic Role Labeling*. Computational Linguistics, 2008

S. Pradhan, W. Ward and J. H. Martin. *Towards Robust Semantic Role Labeling*. Computational Linguistics, 2008

M. Surdeanu and J. Turmo. 2005. *Semantic role labeling using complete syntactic analysis*. CoNLL-2005 shared task, 2005.

K. Toutanova, A. Haghighi and C. Manning. *A Global Joint Model for Semantic Role Labeling*. Computational Linguistics, 2008

Joint Parsing of Syntactic and Semantic Dependencies. http://barcelona.research.yahoo.net/dokuwiki/doku.php?id=conll2008:description, 2008

Dekai Wu and Pascale Fung. *Semantic roles for smt: A hybrid two-pass model*. NAACL-HLT, 2009

S. Wu, J. D. Choi and M. Palmer. *Detecting cross-lingual semantic similarity using parallel propbanks*. AMTA, 2010.

S. Wu and M. Palmer. *Semantic Mapping Using Automatic Word Alignment and Semantic Role Labeling*. ACL-SSST5, 2011