Natural Language Applications for Decision Trees

- Syntactic parsing (Magerman 1995; Haruno et al. 1999)
- Noun phrase coreference (Aone & Bennett, 1995; McCarthy & Lehnert, 1995)
- Cue phrase identification in text and speech (Litman, 1994; Siegel & McKeown, 1994)
- Discourse structure classification from intonational features (Grosz & Hirschberg, 1992)
- Discourse analysis in information extraction (Soderland & Lehnert 1994)
- Lexical tagging: part-of-speech, semantic classes (Cardie, 1993)
- Word sense disambiguation (Mooney, 1996)

Noun Phrase Coreference

[John Simon], [chief Financial Officier] of [Prime Corp.] since 1986, saw [his] pay jump 20%, to $1.3 million, as [financial services company]'s [the 37-year-old] also became [the president].

- MLR (Aone & Bennett, 1995)
- RESOLVE (McCarthy & Lehnert, 1995)
- Corpus
  - texts annotated with coreference information
  - links refer to the most recent coreferent NP
- Training instance creation
  - convert coreference to a classification task
  - one instance for each pair of possible referents
  - features: describe each phrase in isolation as well as relationships between the phrases
  - class value: coref, not-coref

$$\underbrace{p1_1,\ p1_2,\ \ldots,\ p1_n,}_{\text{phrase 1}}\ \underbrace{p2_1,\ p2_2,\ \ldots,\ p2_n,}_{\text{phrase 2}}\ \underbrace{c_1,\ c_2,\ \ldots,\ c_n,}_{\text{relations}}\ \underbrace{C}_{\text{class}}$$

MLR Training

- Training instances
  - 66 features per instance, e.g.
    - semantic class of head noun
    - lexical category of head noun
    - topicalization
    - grammatical category

    phrase features

    ---

    relation features

    - number
    - phrase1 precedes phrase2
    - phrase1 subsequence of phrase2
    - phrase1 topicalization matches phrase2
    - phrase1 semantic class subsumes phrase2

- Generating the training set
  - For each NP anaphor in the training text, generate
    - positive examples for all antecedents on the anaphoric chain

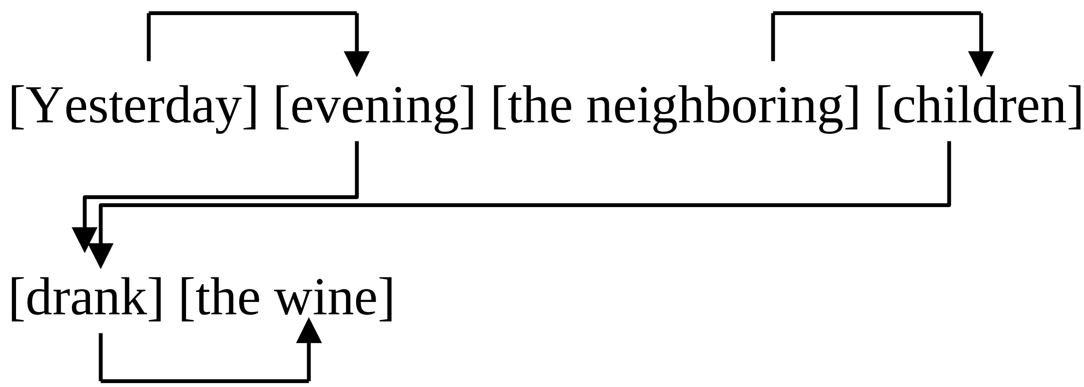
      C -> B -> A: C-B, C-A, B-A
    - negative examples for all possible antecedents not in the anaphoric chain
  - 1971 anaphora from 295 texts

MLR Results

- Testing
  - 1359 anaphora from 200 blind texts
  - decision tree may predict more than one antecedent for a given anaphor
- Scoring
  - recall = # correct / # anaphora in answer key
  - precision # correct / # resolutions attempted
- Results
  - 69.7R / 86.7P
  - results are somewhat inflated because anaphora with multiple, discontinuous referents are omitted and because only anaphora identified by their NLP system are considered
  - manually designed anaphora resolution system (Aone & McKee, 1993)
    - 66.5R / 72.9 P

Japanese Dependency Parser
(Haruno et al., 1999)

- Segment a sentence into a sequence of bunsetsu.



- Prepare a modification matrix, each value of which represents how likely one bunsetsu is to modify another.

	yesterday		
evening	0.70	*evening*	
the neighboring	0.07	0.10	*the neighboring*
children	0.10	0.10	0.70
...			

- Find optimal modifications in a sentence by dynamic programming.
  - Standard bottom-up chart parsing.

Decision Trees for Modification Matrix Construction

- Notation

  Sentence S comprises a set of bunsetsu B,

  $$S = B = \{b_1,\ldots,b_m\}$$

  Define D to be a modification set,

  $$D = \{\ \mathit{mod}(1),\ \ldots\ ,\ \mathit{mod}(m-1)\ \}$$

  where *mod(i)* is the bunsetsu that is modified by the *i*th bunsetsu.

- Parser goal
  - find optimal D

- Decision tree goal
  - produce entries in modification matrix

Feature Set

- 13 features
- 5 each for bunsetsu b_i, b_j
  - part-of-speech of head word
  - type of bunsetsu
  - punctuation
  - parentheses
  - lexical information of head word
    - frequent word
    - thesaurus category
- 3 for relationship between b_i and b_j
  - distance between two bunsetsu
    - none
    - between 1 and 4
    - 5 or more
  - particle 'wa' between two bunsetsu
  - punctuation between two bunsetsu
- Class
  - *yes*: b_i modifies b_j
  - *no*: b_i does not modify b_j

Changes to Decision Tree Algorithm

- Parser assigns most plausible modification set D_{best} to a sentence:

$$D_{best} = argmax_D \, P\,(D \mid B)$$

- Adopt an independence assumption:

$$P\,(D \mid B) = \prod_{i=1}^{m-1} P\,(yes \mid b_i, b_j, f_{ij})$$

- Modify decision tree for regression rather than classification. Extract class frequencies at every node in the decision tree.

- Compute *P (yes)* at every node.

$$P\,(yes \mid b_i, b_j, f_{ij}) = \frac{P_{DT}\,(yes \mid b_i, b_j, f_{ij})}{\sum_{k>i}^{m} P_{DT}\,(yes \mid b_i, b_k\, f_{ik})}$$

Evaluation

- EDR Japanese annotated corpus
- Did not use lexical information feature
- Training: 50,000 sentences
- Testing: 10,000 sentences
- Only used sentences with correct bunsetsu segmentation.
- Accuracy = Precision
  - 84.33%P
  - beats the best stochastic parser for Japanese
  - close to best stochastic English parsers
    - 86-87%P

Qualitative Evaluation

- Investigated importance of individual features
  - bunsetsu type and distance most important
- Some advantages over Collins-style stochastic dependency parser (Collins, 1996 and 1997)
  - Collins defines a set of attributes and conditions the modification probabilities for all attachment decisions on all attributes regardless of the bunsetsu type.
  - Collins can include only a small number of features due to sparse data problems.
- Haruno et al.'s approach allows the use of an arbitrary number of attributes.
- Decision trees allow a more sophisticated modification matrix than traditional methods. Selects sufficient number of significant attributes according to bunsetsu type.