

Widening the ACE Pipeline

Edward Loper



1

CIS-630

The U. Penn Pipeline

- The U. Penn ACE system uses a *pipeline* of independent *stages* to process text.
 - Tokenizers (sentence & word)
 - Tagger
 - Parser
 - Named Entity Classifier
 - Nominal Classifier
 - Coreference Classifier
 - Relation Classifier



2

CIS-630

Pipelines

- Advantages of a pipelined system:
 - Stages can be optimized independantly
 - Different implementations of each stage can be easily integrated into the system
 - Easy to use existing components
- Disadvantages of a pipelined system:
 - Interaction between stages is limited
 - If one stage produces incorrect output, later stages can't usually recover.



3

CIS-630

Wide Pipelines

A *wide pipeline* attempts to keep the advantages of a pipelined system, while avoiding the shortcomings.

- Instead of passing a single value from each stage to the next, pass multiple values.
- A simple approach: each stage produces the *n-best* outputs (with probabilities).
- Problems with the n-best approach:
 - Exponential growth of outputs.
 - Poor coverage.



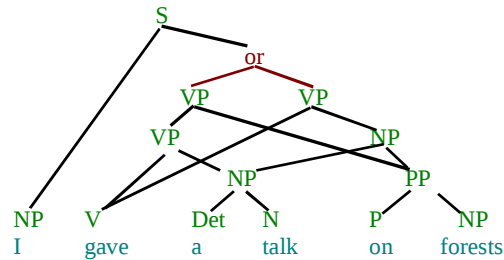
4

CIS-630

Shared Forests

Shared forests provide a compact representation for many outcomes with shared structure

- Use **or** nodes to represent choices.

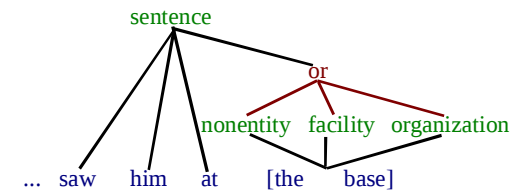


5

CIS-630

Shared Forests: Not Just for Parses

- Shared forests can be used to represent any collection of related outputs with shared substructure.



6

CIS-630

Probabilistic Shared Forests

- Symbolic shared forests are a compact representation for related outputs.
- But probabilistic shared forests impose some constraints on the probability distribution of the outputs.
 - E.g., there's no easy way to block a single output without blocking other outputs that it shares structure with.
 - These constraints can always be avoided by restructuring the tree; but this can be expensive.

7

CIS-630

Project Outline

1. **Decide on a suitable representation.**
 - If possible, maintain compatibility with Tom's framework.
2. **Adapt the U. Penn ACE system to use a wider pipeline.**
 - Minimally, adapt the tokenizer, tagger, parser, and nominal classifier.
3. **Evaluation.**
 - Compare the best outcomes produced by the "wide pipeline" system to the outcomes generated by the current system.

8

CIS-630