Tree Edit Distance for Recognizing Textual Entailment:

Estimating the Cost of Insertion

Milen Kouylekov & Bernardo Magnini

University of Trento,
ITC-irst, Centro per la Ricerca Scientifica e Tecnologica,
Trento, Italy

Venezia, April 10, 2005

Our Goals

- Continue the development of a system based on **Tree Edit Distance**
  - Investigate the cost of **Insertion**
- Combining different system settings using learning algorithm

The Textual Entailment Framework

- General Framework proposed Dagan and Glickman (2004) addressing language variability.
  - An **Entailment Relation** holds between two text fragments (i.e. **text T** and **hypothesis H**) when the meaning of H, as interpreted in the context of T, can be inferred from the meaning of T.
  - **Entailment Rules (patterns)** is directional relation between two parse sub-trees with variables, where the first one entails the second.

Entailment and Tree Edit Distance

- Represent T and H as **dependency trees**
- The probability of an entailment relation between T and H is related to the mappings between H and T
- Mappings can be described as a sequence of **editing operations** that transform T into H
- Each edit operation has a **cost** assigned to it
- Entailment holds if the overall **transformation cost** is below a certain threshold, estimated over the training data.

Tree Edit Distance on Dependency Trees

- (Zhang and Shasha, 1990) Tree Edit Distance algorithm has been implemented.
- Edit operations (Insertion, Deletion, Substitution) are allowed on single nodes only
- Parsing is performed with Minipar (Lin 1998)
- Node order is relevant: node are re-arranged according to: *subj--> obj --> mods*
- The original algorithm does not consider labels on edges: relations names are concatenated to node names
  - E.g. eat [subj] John   eat --> John#subj

Cost functions

- Insertion: the cost of inserting a node *w* in T should be proportional to the relevance of *w* in the context of H.

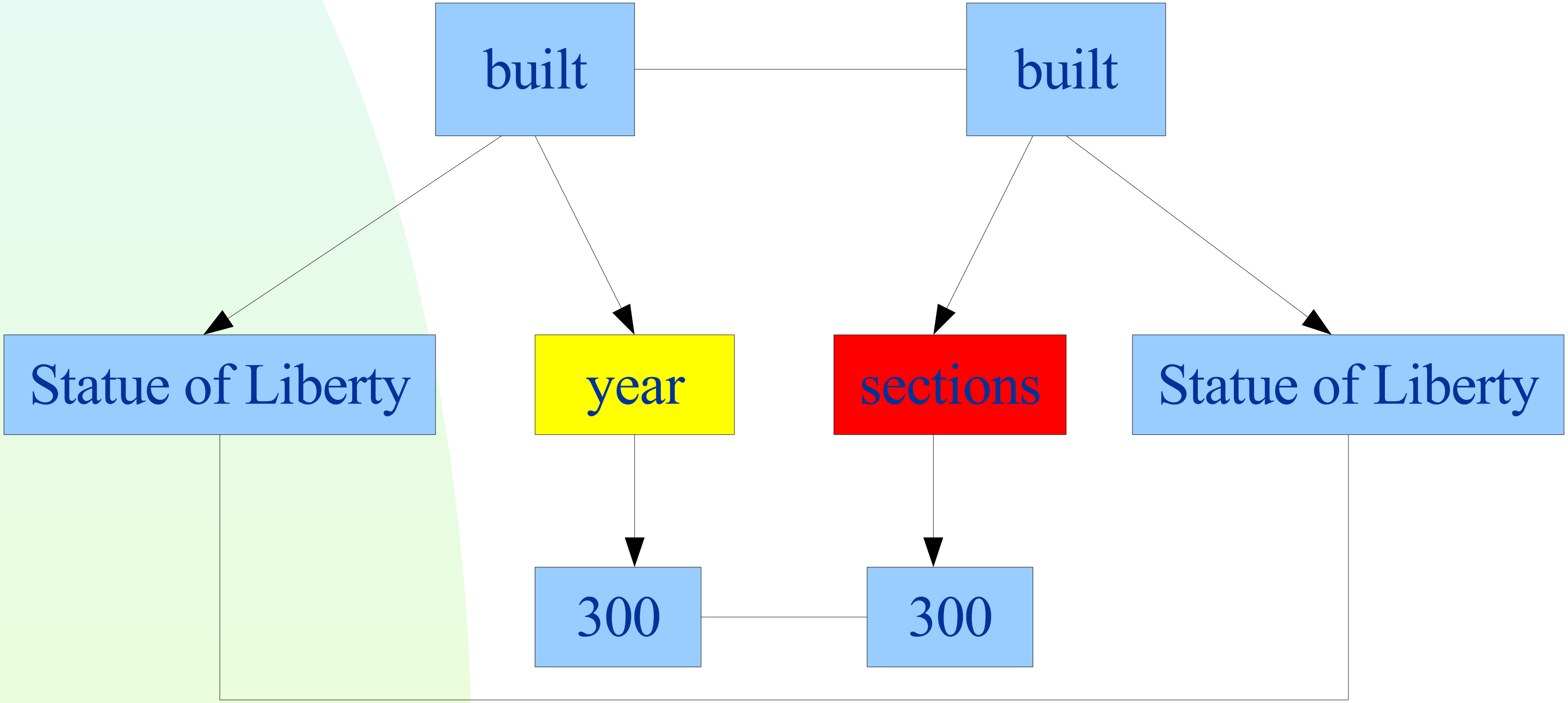
- Deletion: the cost of deleting a node *w* in T should be proportional to the relevance of *w* in the context of T.

- Substitution: the cost of substituting a node *w1#rel1* in T with a node *w2#rel2* in H is proportional to the strength of the entailment relation between the two nodes and relevant to the context of H and T.

Example

T: The Statue of Liberty is so big it had to be built in 300 sections.

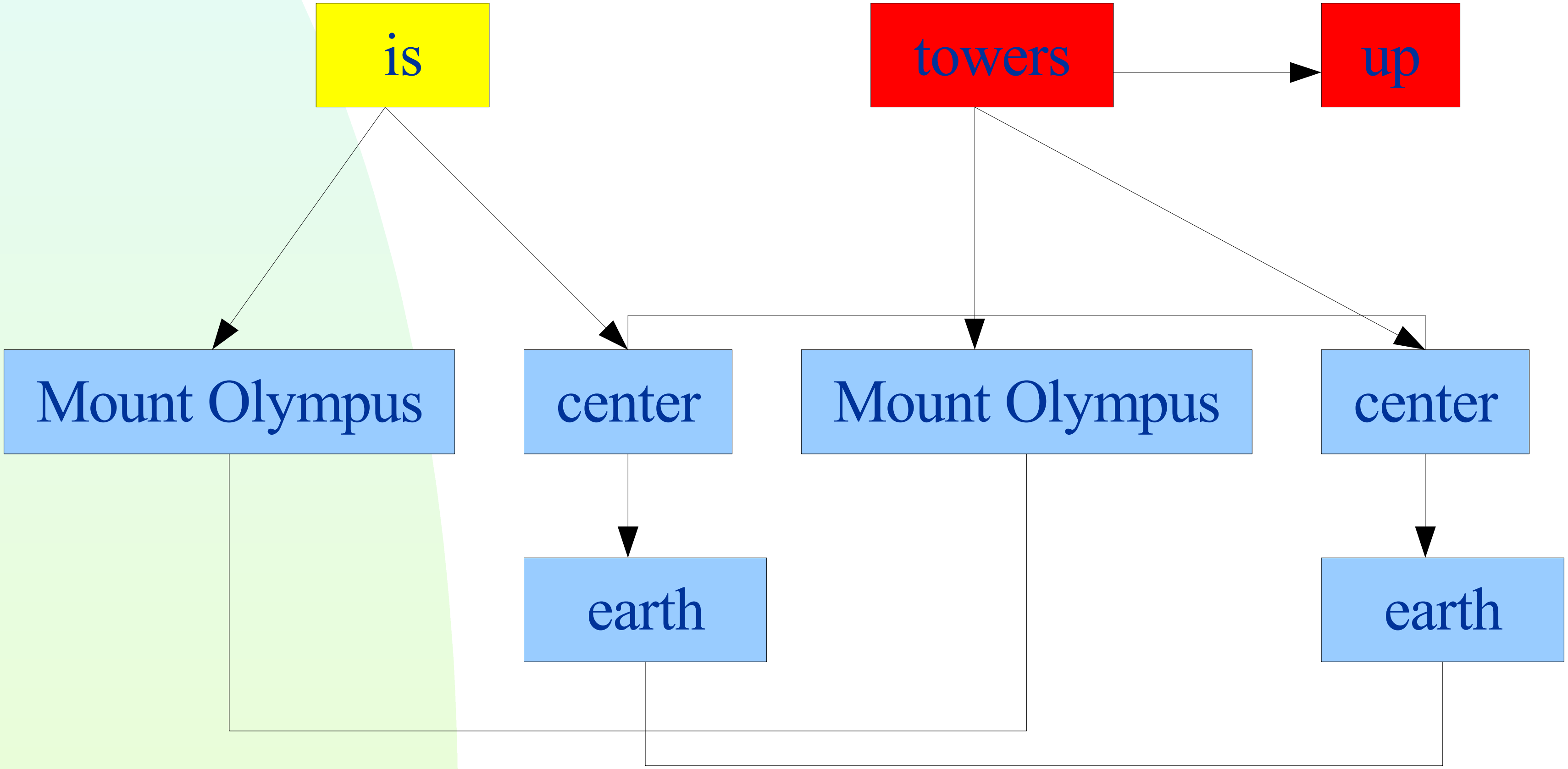
H: The Statue of Liberty was built in the year 300.



Example (2)

T: Mount Olympus towers up from the center of the earth.

H: Mount Olympus is in the center of the earth.



System Settings

- System 1: Insertion as IDF
- System 2: Fixed Insert cost
- System 3: Number of Parents.
- System 4: Number of Children.
- System 5: Number of Children + Number of Parents
- System 6: Combined
  - All previous systems as features of the sequential minimal optimization (SMO) algorithm – training a support vector classifier

System Settings - Performance

	development	cross-validation	test
idf	0.581	0.578	0.572
fixed	0.591	0.560	0.570
#parents	0.600	0.590	**0.582**
#children	0.579	0.579	0.541
#ch + #par	0.598	0.590	0.571
combined	0.637	0.613	**0.605**

- Combined run is the best performing
- Additional resources for calculating the insertion cost are not needed

System Settings - Performance

		IE	IR	QA	SUM	Total
idf	accuracy	**0.5050**	0.5500	0.5650	0.6700	**0.5725**
	precision	0.5095	0.4658	0.4658	0.7067	0.5249
combined	accuracy	**0.5200**	0.6000	0.6000	0.7000	**0.6050**
	precision	0.4978	0.5352	0.5352	0.5240	0.5046

- Our system performs well on the Summarization task
- IE requires a large resource of complex entailment rules.
- Combined run is accurate but less precise.

Thank You!