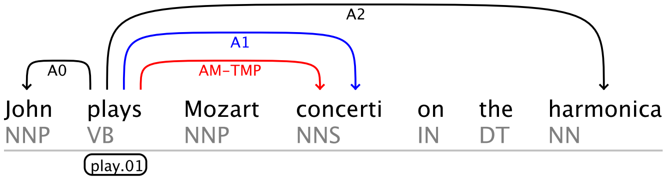
# Collective Semantic Role Labelling with Markov Logic

Sebastian Riedel    Ivan Meza-Ruiz

Institute for Communicating and Collaborative Systems
School of Informatics
University of Edinburgh
{S.R.Riedel,I.V.Meza-Ruiz}@sms.ed.ac.uk

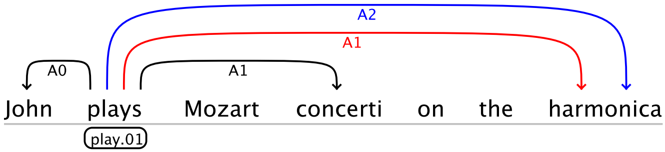
August 16th, 2008

# SRL Intuitions: Local



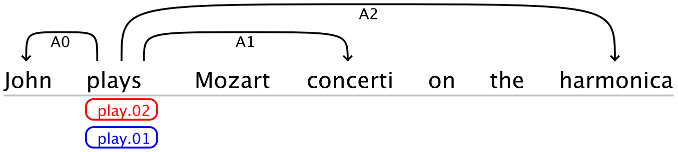

- Semantic role label of a token correlates with its POS tag

# SRL Intuitions: Global




- A predicate can have at most one semantic argument of each type.

# SRL Intuitions: Global And Soft




- Sense of a verb correlates with the roles of its arguments/modifiers.

# Capturing intuitions

How do we capture such intuitions?

# Capturing intuitions

How do we capture such intuitions?

- Local: off-the-shelf classifier, easy to do, (not much Machine Learning expertise necessary)

# Capturing intuitions

How do we capture such intuitions?

- Local: off-the-shelf classifier, easy to do, (not much Machine Learning expertise necessary)
- Global: pipeline, reranking, ILP, approximate heuristic search, harder to engineer, (more Machine Learning expertise necessary.)

# Capturing intuitions

How do we capture such intuitions?

- Local: off-the-shelf classifier, easy to do, (not much Machine Learning expertise necessary)
- Global: pipeline, reranking, ILP, approximate heuristic search, harder to engineer, (more Machine Learning expertise necessary.)

Markov Logic (generally: Statistical relational learning):

- capture local and global intuition using weighted first order formula
- run off-the-shelf training and inference software

# Overview

1. Markov Logic
2. Modelling
3. Experiments
4. Conclusion

# Markov Logic [Richardson & Pedro (2005)]

What?

- Combines of FOL and Markov Networks
- Defines a log-linear distribution over possible worlds
- Uses weighted FOL formulae

Why?

- Decoupling: fosters independent progress in Machine Learning and application domains
- Induction: automatically induce new rules
- Efficiency: first order representation can be exploited in inference and learning

# Vocabulary

The vocabulary consists of:

- **Constants** represent objects of the domain (e.g., Haag, VB, 1, 2, 3, . . . )
- **Predicates** represent relations over the objects

There are two types of predicates: observable and hidden.
Some of the observable predicates are:

- **word/2**, *word(1,Haag)*
- **pos/2**, *pos(1,NNP)*
- **path/3**, *path(2,1,↑↑↓)*

# Hidden predicates

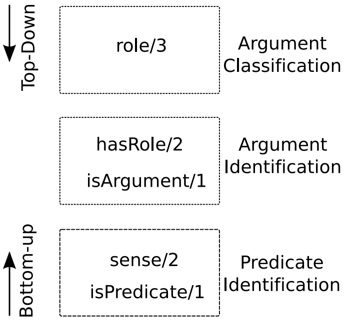



Figure: Hidden predicates

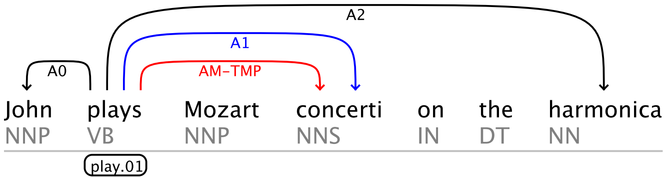
Input :

$$\{word(1, Mr.), word(2, Haag), word(3, plays), word(4, elianti), \ldots\}$$

Output :

$$\{isPredicate(3), sense(3, 02), isArg(2), hasRole(3, 2), role(3, 2, A0), \ldots\}$$

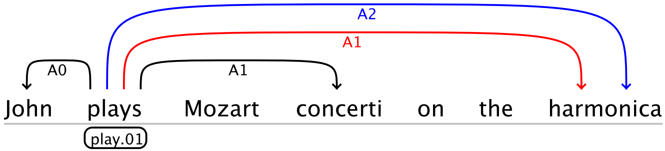
# Formulae: Local




- Semantic role label of a token correlates with its POS tag

$$ppos(p, +p_p) \wedge ppos(a, +p_a) \Rightarrow role(p, a, +r)$$

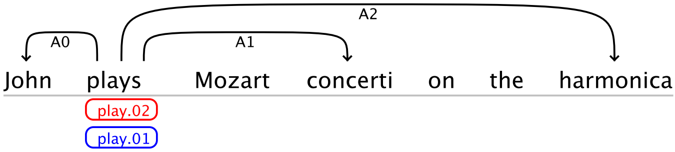
# Formulae: Global




- A predicate can have at most one semantic argument of each type.

$$role(p, a_1, r) \wedge \neg mod(r) \wedge a_1 \neq a_2 \Rightarrow$$
$$\neg role(p, a_2, r)$$

# Formulae: Global and Soft




- Sense of a verb correlates with the roles of its arguments/modifiers.

$$lemma(p, +l) \wedge ppos(a, +s)$$
$$\wedge hasRole(p, a) \Rightarrow sense(p, +f)$$

# Formulae: Summary

Implemented:

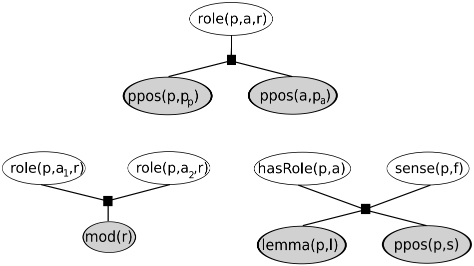
- Local formulae [Xue & Palmer (2004)]
- Global formulae for one predicate
- Global formulae for two different predicates (Structural formulae)

$$isPredicate/1 \leftrightarrow sense/2$$
$$isPredicate/1 \leftrightarrow isArgument/1$$
$$isArgument/1 \leftrightarrow hasRole/2$$
$$hasRole/2 \leftrightarrow role/3$$

# Markov Logic Network

Collect weighted formulae into a **Markov Logic Network** (MLN).
An MLN defines a Markov Network where:

- There is a binary node for each ground atom (e.g., *role(2,1,A1)*.)
- There is a factor for each assignment of the free variable of each formulae.



# Experiments

- Full model: includes all the rules we came up with.
- Bottom-Up model: discard top-down rules, it resembles a pipeline where candidates are picked by latter module.
- Top-down model: discard bottom-up rules, it resembles a pipeline where candidates are picked by earlier module.
- Isolated: discard bottom-up and top-down rules.
- Structural: discard the soft global rules.

# Results

| Model | WSJ | Brown | Train Time | Test Time |
|---|---|---|---|---|
| Full | 75.72% | **65.38**% | 25h | 24m |
| Up | **76.96**% | 63.86% | 11h | 14m |
| Down | 73.48% | 59.34% | 22h | 23m |
| Isolated | 60.49% | 48.12% | 11h | 14m |
| Structural | 74.93% | 64.23% | 22h | 33m |

Table: F-scores for different models.

# Conclusions

- This network achieves the second best semantic F- scores in the Open Track of the CoNLL shared task for only SRL.
- The bottom-up model outperforms the full model on WSJ.
- ML: minimal engineering efforts + state-of-the-art results.

# Pointers

- Implementation of ML:
  `http://thebeast.googlecode.com/`
- Models and scripts:
  `http://thebeast.googlecode.com/svn/mlns/conll08/`