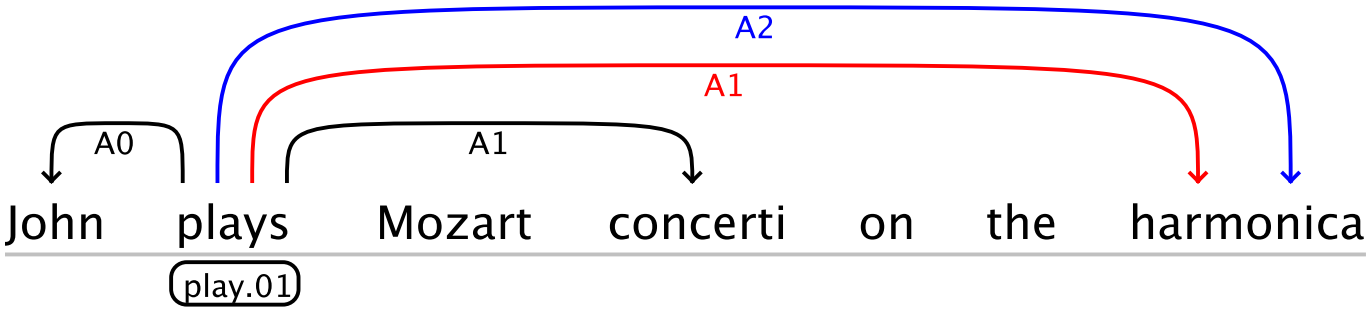
Collective Semantic Role Labelling with Markov Logic

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

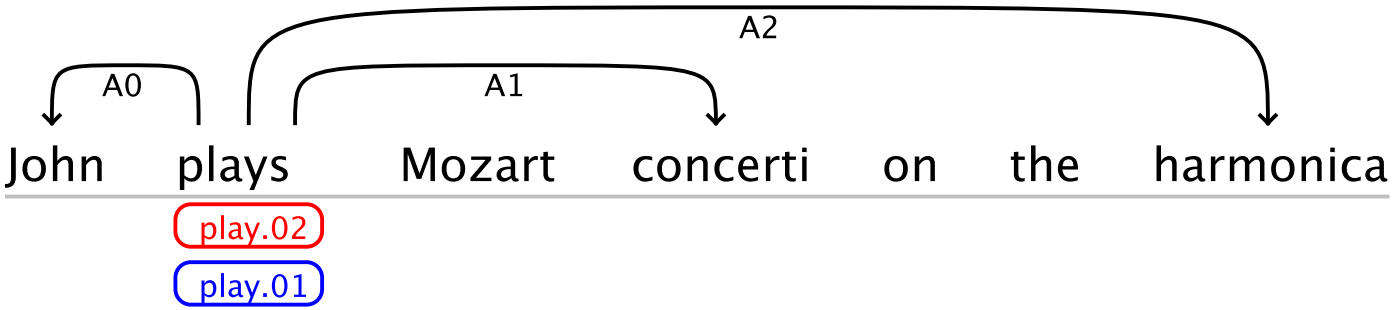
August 16th, 2008

Our intuitions of the SRL task



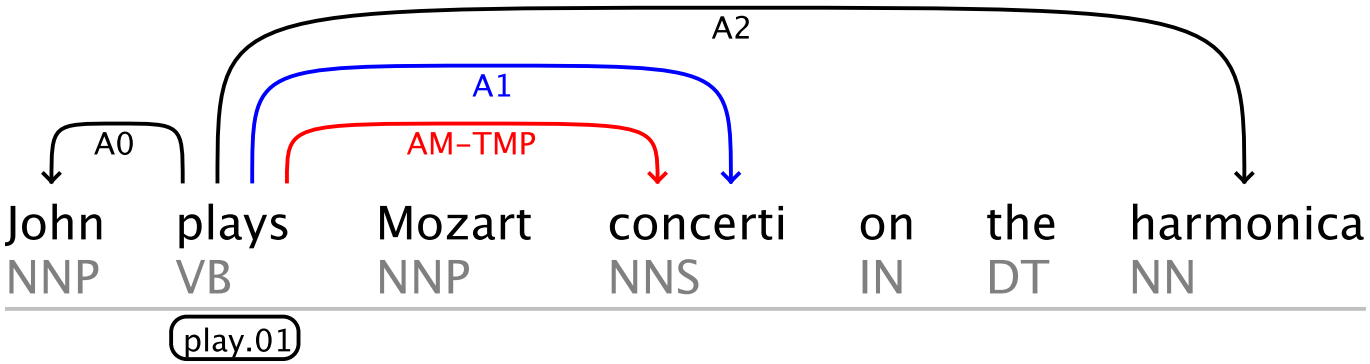
- A predicate can have at most one argument of a proper argument role.

Our intuitions of the SRL task



- Sense of a verbe correlates with the arguments roles of a verb.

Our intuitions of the SRL task



- Semantic role labels correlates with their POS tags

Capturing our intuitions

- Local: classifier, easy to do
- Global: pipeline, reranking, ILP, approximate heuristic search, harder to engineer
- Markov Logic (other, Statistical relational learning), easy to capture intuition

Markov Logic

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

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

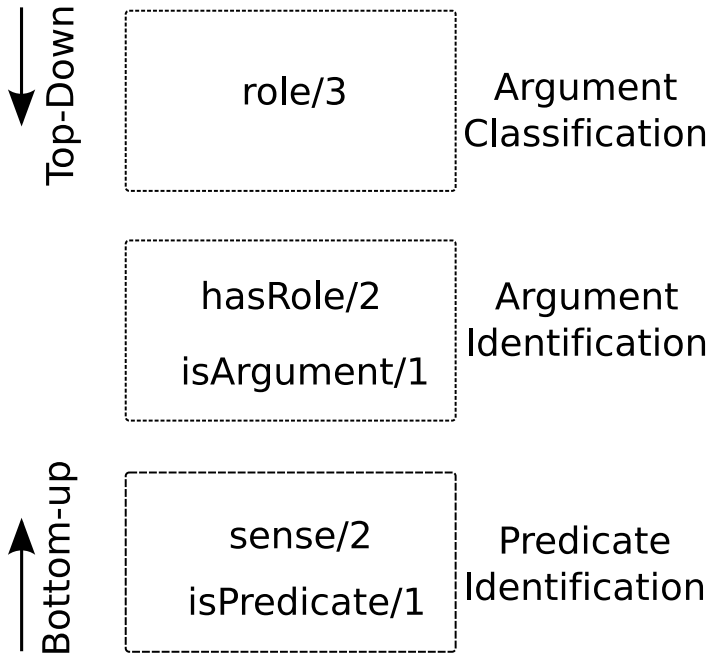


Figure: Hidden predicates

Input :

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

Output :

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

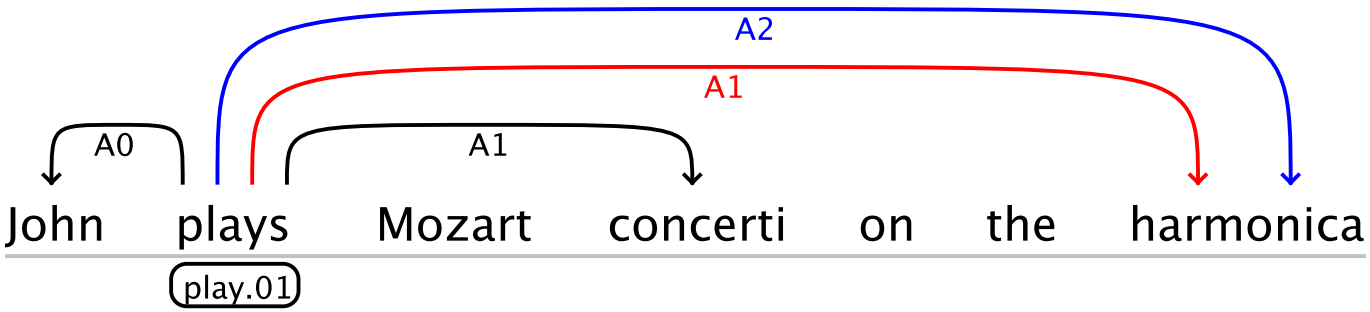
Formulae

We use formulae to capture statements about the world.

- Local formulae
- Global formulae

Formulae

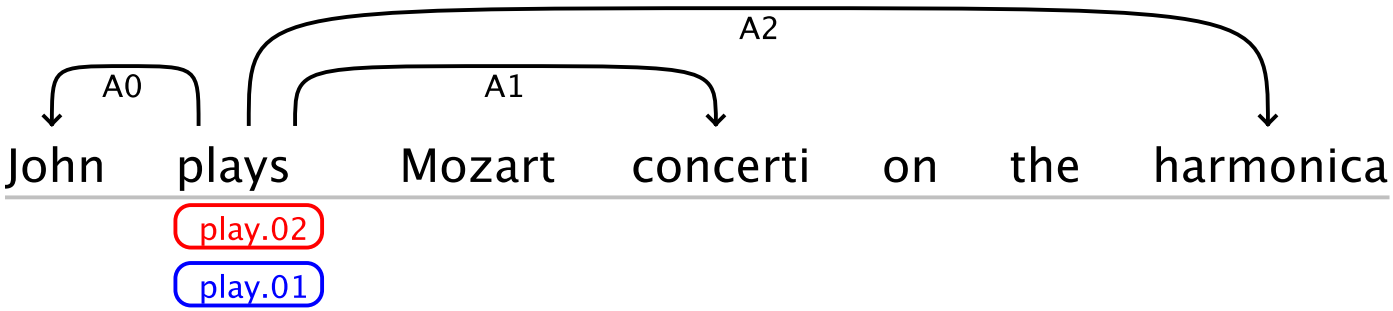
Structural constraints ensure the possible world is valid



$$role(p, a, r_1) \wedge r_1 \neq r_2 \Rightarrow \neg role(p, a, r_2)$$

Formulae

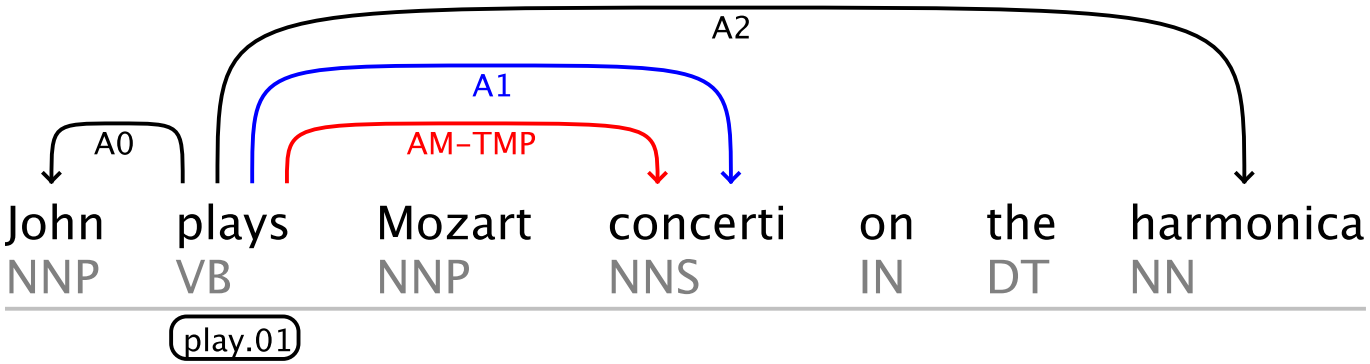
Hard constraints about the nature of the task



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

Formulae

Soft constraints which depend on the configuration of the predicates



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

Markov Logic Network

A set a weighted formulae is call an 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

- Whole model: includes all the rules we come 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 reaches a better performance.

Pointers

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