Joint Learning of Syntactic and Semantic Dependencies

Xavier Lluís and **Lluís Màrquez**
TALP Research Center
Technical University of Catalonia

Barcelona, December 9, 2008

Introduction

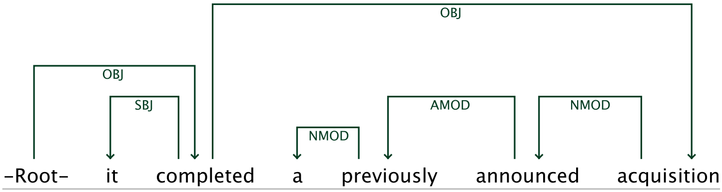
Joint parsing is the **simultaneous** processing of the syntactic and semantic structure.

Syntactic and semantic parsing: syntax

-Root- it completed a previously announced acquisition

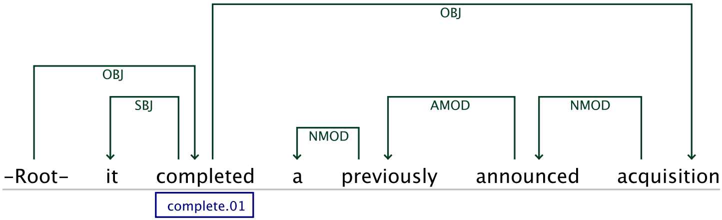
A sample sentence

Syntactic and semantic parsing: syntax



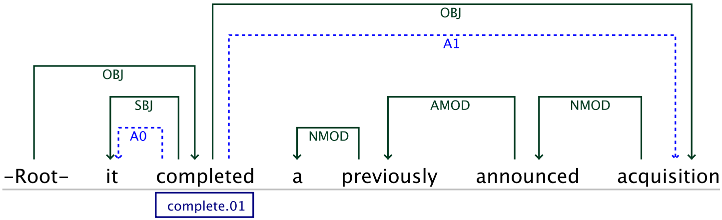
Syntactic dependencies

Syntactic and semantic parsing: semantics



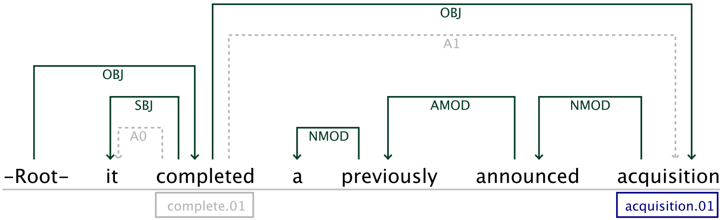
Predicate completed

Syntactic and semantic parsing: semantics



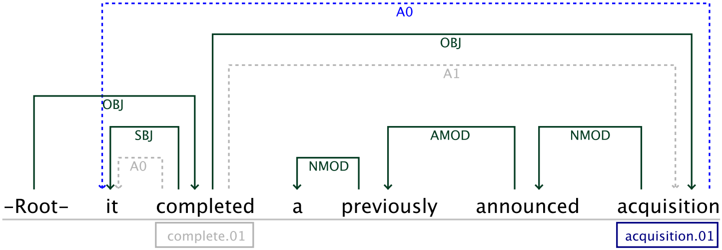
Semantic dependencies for completed

Syntactic and semantic parsing: semantics



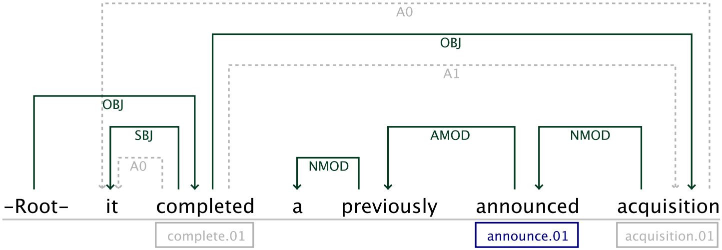
Predicate acquisition

Syntactic and semantic parsing: semantics



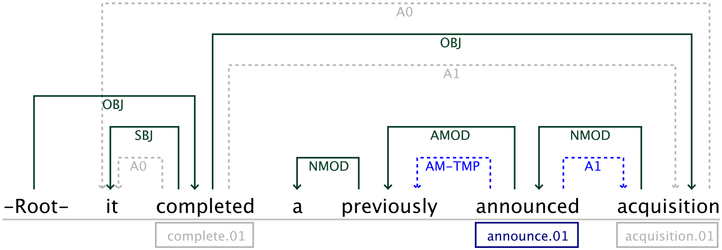
Semantic dependencies for acquisition

Syntactic and semantic parsing: semantics



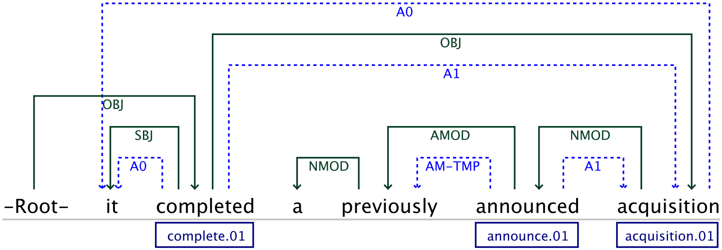
Predicate announced

Syntactic and semantic parsing: semantics



Semantic dependencies for announced

Syntactic and semantic parsing: semantics

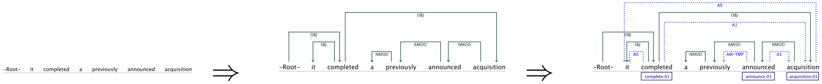


Semantic dependencies for all predicates

Mainstream approach

The pipeline approach

1. Syntactic parsing
    - A parser (Eisner, Shift-reduce)
2. Semantic role labeling
    - A simpler (non-structured) classifier



Pipeline strategy

The pipeline approach

1. Propagation or amplification of errors
2. Assumes an order of increasing difficulty
3. Dependencies between layers are hard to be captured

Joint approach

Design a joint model

1. Overcome the pipeline approach
2. To build from scratch a simple and feasible system

Design a joint model

A joint approach

Extend a syntactic parsing model to **jointly** parse semantics

1. Syntactic parsing
   - A parser (**Eisner**, Shift-reduce)
2. Semantic role labeling
   - A simpler (non-structured) classifier

Design a joint model

A joint approach

Extend the **Eisner** algorithm to **jointly** parse semantics

- $O(n^3)$ algorithm
- Based on CKY algorithm
- Bottom-up parser

The Eisner algorithm

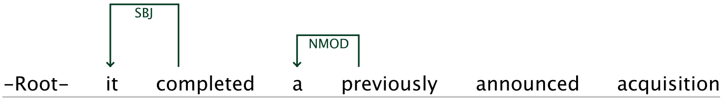
-Root- it completed a previously announced acquisition

Bottom-up dependency parsing

The Eisner algorithm

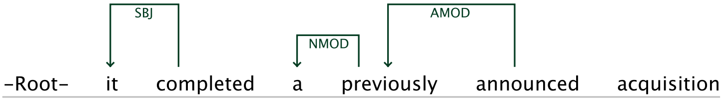
Bottom-up dependency parsing

The Eisner algorithm



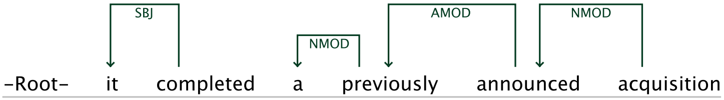
Bottom-up dependency parsing

The Eisner algorithm



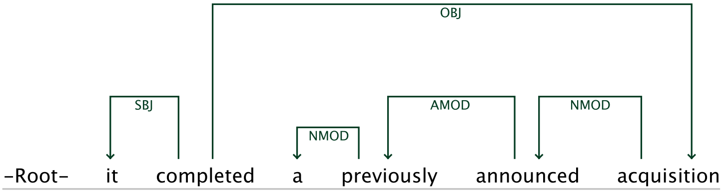
Bottom-up dependency parsing

The Eisner algorithm



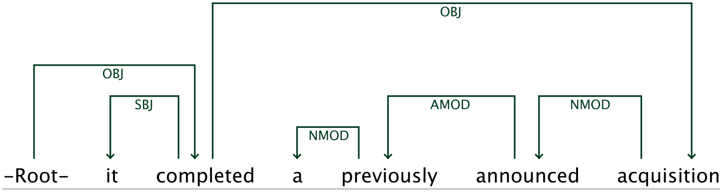
Bottom-up dependency parsing

The Eisner algorithm



Bottom-up dependency parsing

The Eisner algorithm



Bottom-up dependency parsing

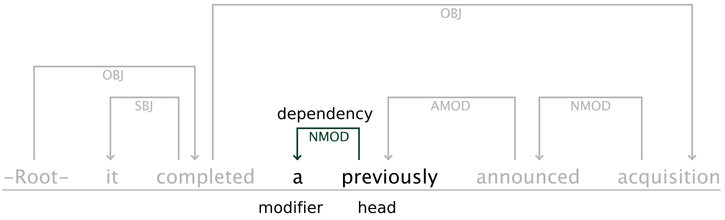
The Eisner algorithm

Score of a dependency

A dependency $d = \langle h, m, l \rangle$ of a sentence x is scored by:

$$\text{score}(d, x) = \phi(\langle h, m, l \rangle, x) \cdot \mathbf{w}$$

where ϕ is a feature extraction function,
$\mathbf{w}$ is a weight vector



The Eisner algorithm

Best tree

We are interested in the best scoring tree among all trees $\mathcal{Y}(x)$:

$$\text{best\_tree}(x) = \underset{y \in \mathcal{Y}(x)}{\operatorname{argmax}} \, \text{score\_tree}(y, x)$$

Eisner algorithm

The Eisner algorithm is an **exact** search algorithm that computes the best first-order factorized tree.

The Eisner algorithm

Score of a tree

A syntactic tree y for a sentence x is scored by:

$$\text{score\_tree}(y, x) = \sum_{\langle h,m,l\rangle \in y} \text{score}\left(\langle h, m, l\rangle, y\right)$$

Arc-factorization

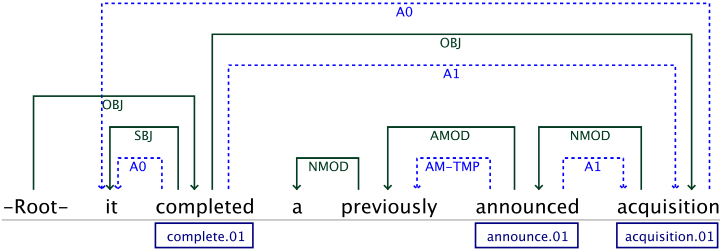
The first order factorization is the sum of independent scores for each dependency of the tree.

Extension of the Eisner algorithm

Joint parsing point of view

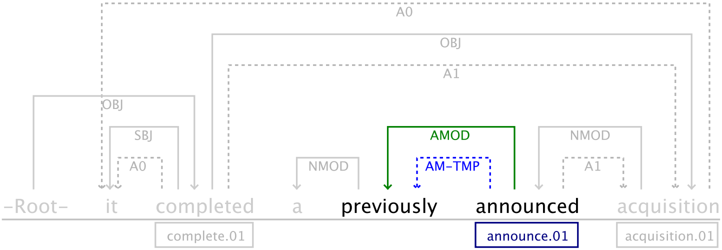
simultaneous prediction of the syntactic and semantic label

Extension of the Eisner algorithm: an example



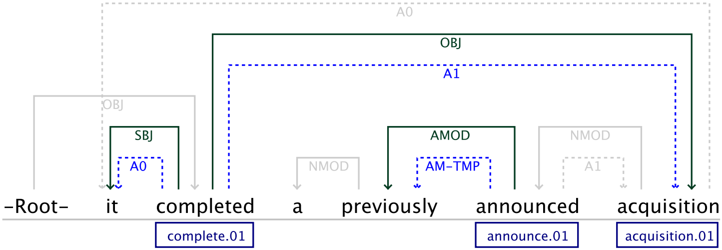
The complete syntactic and semantic structure.

Extension of the Eisner algorithm: an example



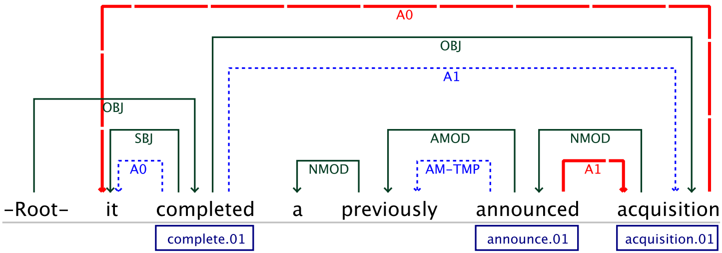
Overlapping syntactic and semantic depencies.

Extension of the Eisner algorithm: an example



Overlapping syntactic and semantic depencies.

Extension of the Eisner algorithm: an example



Non-overlapping semantic dependencies.

Syntax and Semantics overlapping

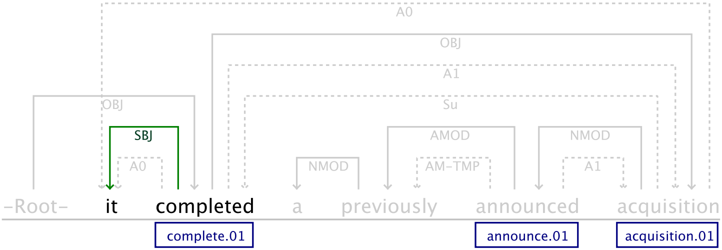
1. Are syntax and semantics overlapping?

- **36.4%** of argument-predicate relations do not exactly overlap with modifier-head syntactic relations.

Proposed solution

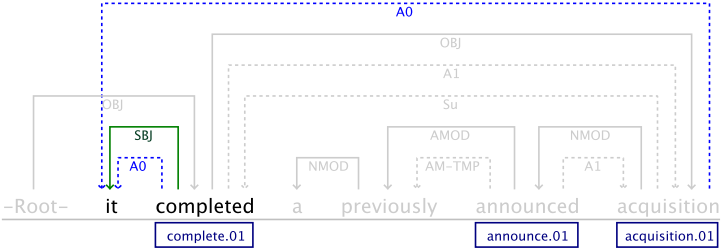
Attach the semantic label to the syntactic dependency

Difficulties: non-overlapping semantics



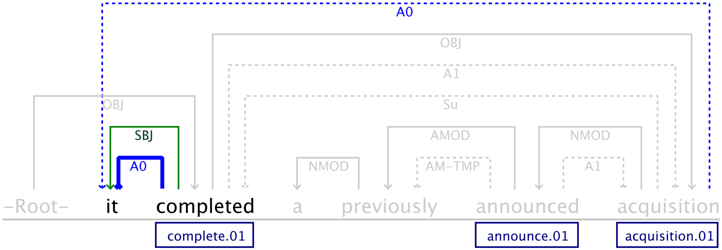
Any given syntactic dependency

Difficulties: non-overlapping semantics



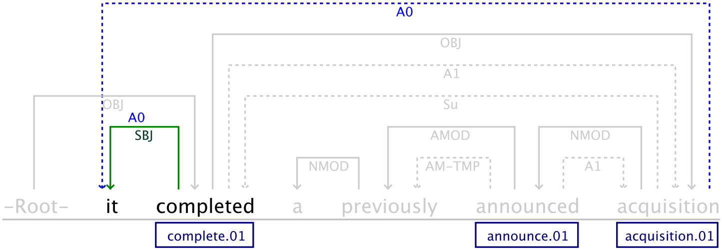
The related semantic dependencies

Difficulties: non-overlapping semantics



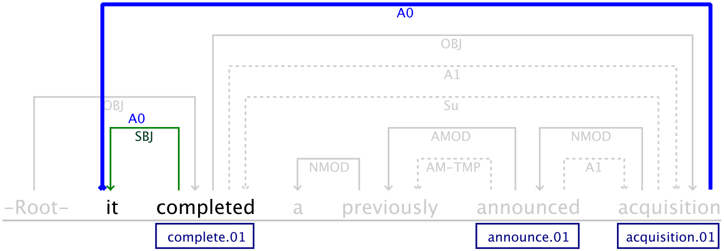
The overlapping A0 dependency

Difficulties: non-overlapping semantics



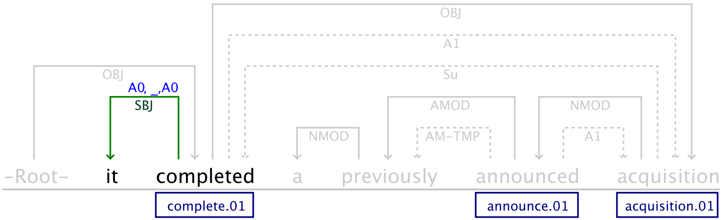
The overlapping A0 dependency will be jointly annotated

Difficulties: non-overlapping semantics



The non-overlapping A0 dependency

Difficulties: non-overlapping semantics



The non-overlapping A0 dependency will also be jointly annotated

Difficulties: non-overlapping semantics

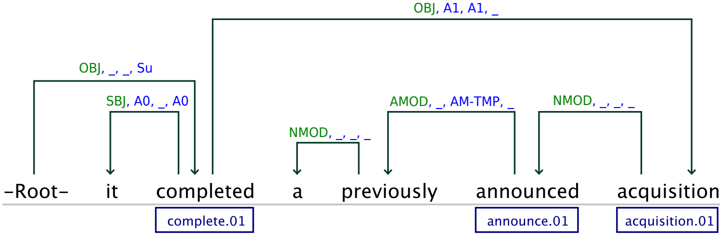
Solution

An extended dependency is:

$$d = \langle h, m, l_{syn}, l_{sem\ p_1}, \ldots, l_{sem\ p_q} \rangle$$

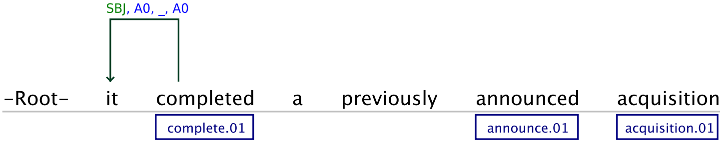
h is the head
m the modifier
l_{syn} the syntactic label
$l_{sem\ p_i}$ one semantic label for each sentence predicate p_i

Proposed solution



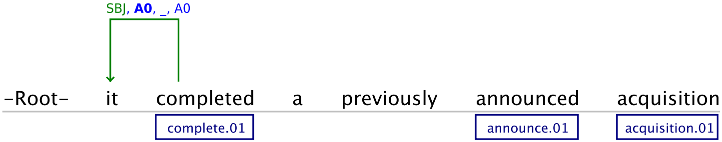
A dependency has its syntactic and semantic labels

Proposed solution: unavailable features



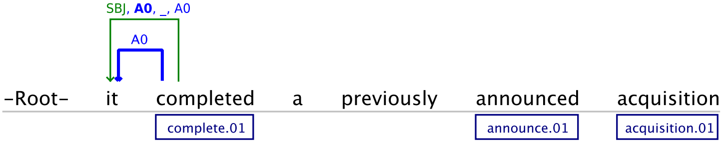
A dependency with semantic labels

Proposed solution: unavailable features



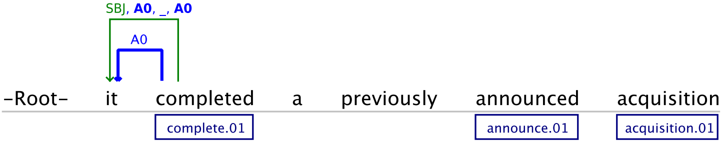
The first A0 is an overlapping semantic dependency

Proposed solution: unavailable features



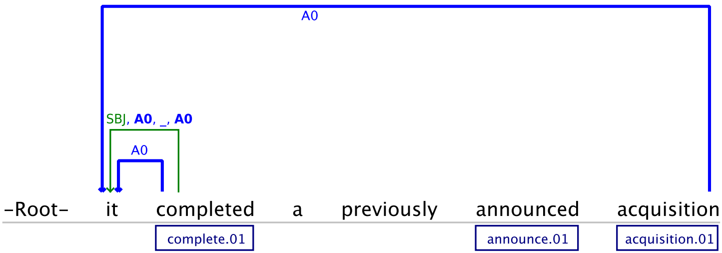
The first A0 is an overlapping semantic dependency

Proposed solution: unavailable features



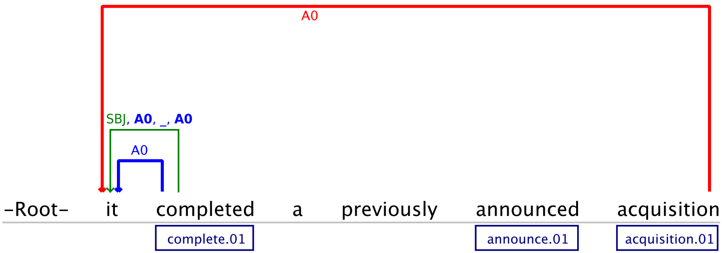
The second A0 is a non-overlapping semantic dependency

Proposed solution: unavailable features



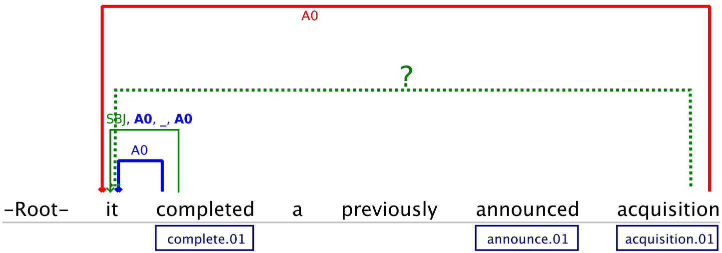
The second A0 is a non-overlapping semantic dependency

Proposed solution: unavailable features



The second A0 is a non-overlapping semantic dependency

Proposed solution: unavailable features



The syntactic relation is not yet processed

Problems inherited from traditional pipeline design

2. Problems not appearing in pipeline systems

- State-of-the-art SRL systems strongly rely on syntactic path features.
- There is only a partial visibility of the syntax restricted to the current sentence span.
- A distant argument-predicate relation can occur.

Proposed solution

Pre-parse and extract predicate-modifier syntactic paths.

Joint Model

Joint model formalization

Joint Model

The Joint Model extends and it is based on the first order syntactic model

Best joint tree

$$\text{best\_tree}(x, \mathbf{w}, y') = \operatorname*{argmax}_{y \in \mathcal{Y}(x)} \text{score\_tree}(y, x, \mathbf{w}, y')$$

argmax computed using the Eisner algorithm

x is the input sentence
y is the syntactic-semantic tree
y' pre-parsed syntactic tree
$\mathbf{w}$ is the weight vector

Joint model

First order factorization

$$\text{score\_tree}(y, x, \mathbf{w}, y') = \sum_{\langle h, m, l_{syn}, \mathbf{l} \rangle \in y} \mathbf{score}(\langle h, m, l_{syn}, \mathbf{l} \rangle, x, \mathbf{w}, y')$$

x is the input sentence
y is the syntactic-semantic tree
y' pre-parsed syntactic tree
$\mathbf{w}$ is the weight vector
$\mathbf{l} = l_{sem\ p_1}, \ldots, l_{sem\ p_q}$ are the semantic labels for predicates p_i

Scoring

$$\textbf{score}\left(\langle h, m, l_{syn}, \mathbf{l}\rangle, x, \mathbf{w}, y'\right) =$$
$$\text{syntactic\_score}\left(h, m, l_{syn}, x, \mathbf{w}\right) +$$
$$\text{semantic\_score}\left(h, m, l_{sem\ p_1}, \ldots, l_{sem\ p_q}, x, \mathbf{w}, y'\right)$$

The score of a dependency is the syntactic score (as usual) + the semantic score of the assigned semantic label (if any) for each predicate

$\mathbf{l} = l_{sem\ p_1}, \ldots, l_{sem\ p_q}$

Semantic Scoring

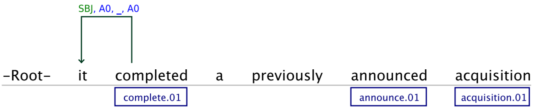
Semantic scoring function

$$\text{semantic\_score}\left(h, m, l_{sem\ p_1}, \ldots, l_{sem\ p_q}, x, \mathbf{w}, y'\right) =$$

$$\sum_{l_{sem\ p_i}} \frac{\phi_{sem}\left(\langle h, m, l_{sem\ p_i}\rangle, p_i, x, y'\right) \cdot \mathbf{w}^{(l_{sem\ p_i})}}{q}$$

y' is the precomputed syntax tree for feature extraction

$l_{sem\ p_i}$ is the semantic label of m for predicate p_i



System summary

Core

Averaged perceptron learning + Eisner algorithm inference
Collins, 2002 *Eisner, 1996*

and based on Carreras et al. 2006

Features

State-of-the-art features adapted to the dependency formalism:

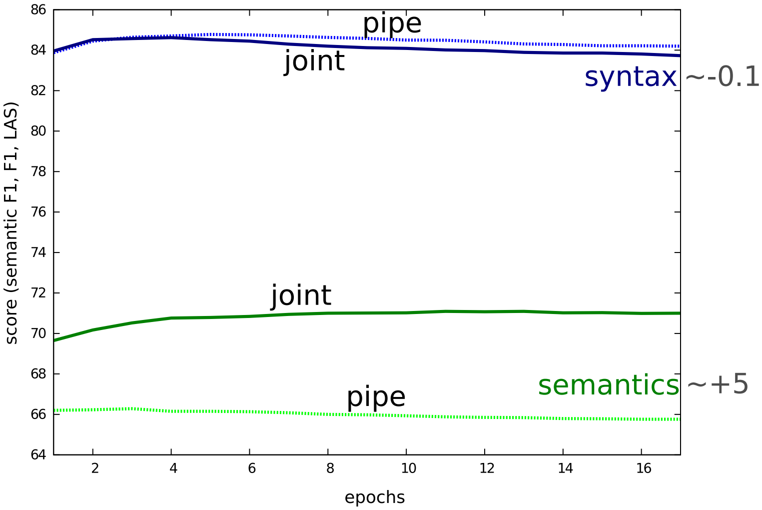
syntax McDonald et al. (2005) and Carreras et al. (2006)

semantics Xue and Palmer (2004) and Surdeanu et al. (2007)

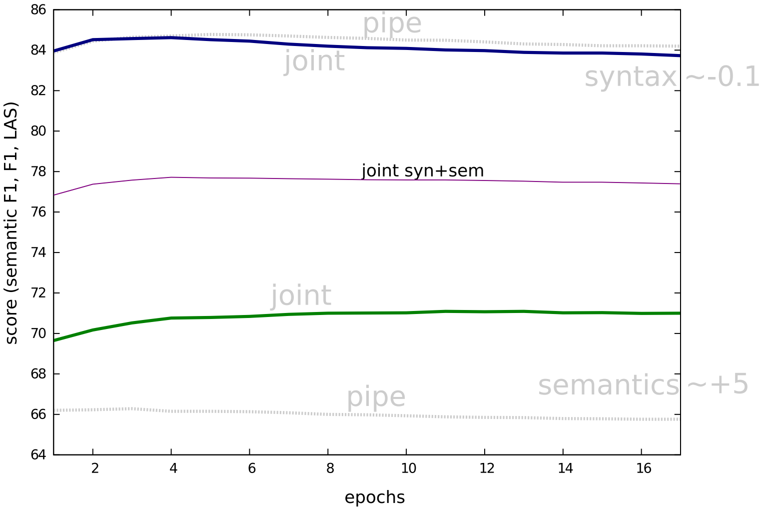
Results

The system was presented to the CoNLL-2008 shared task.

Learning curve (development)



Learning curve (development)

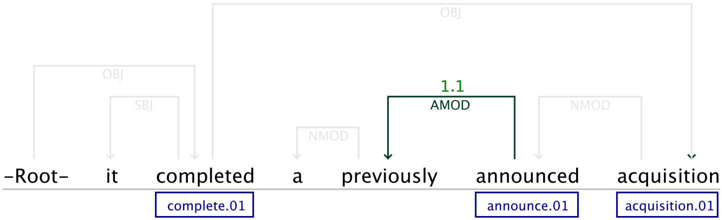


Discussion

Could semantics hurt syntax?

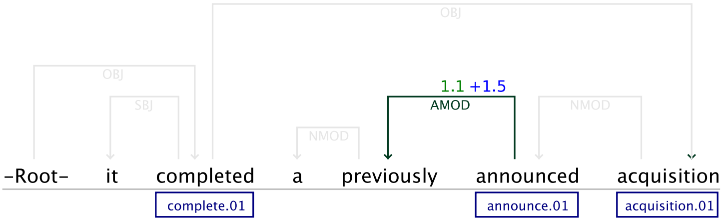
- Analize the effects of semantics $\Rightarrow$ syntax
  - The semantic score increases the overall dependency score
  - The overall dependency score defines the syntax

The syntactic and semantic scores on a dependency



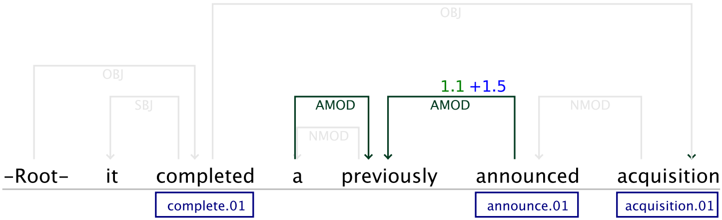
A correct syntactic dependency with its syntactic score

The syntactic and semantic scores on a dependency



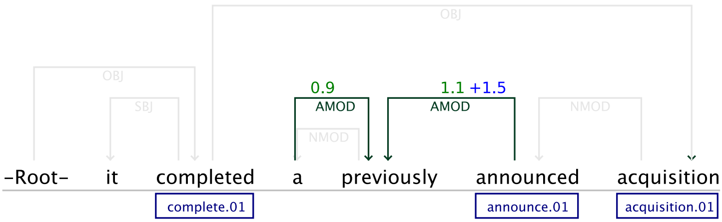
A correct syntactic dependency with its score increased by the semantic score: improved ↑ syntax

What if the semantic score is not so dependant on syntax?



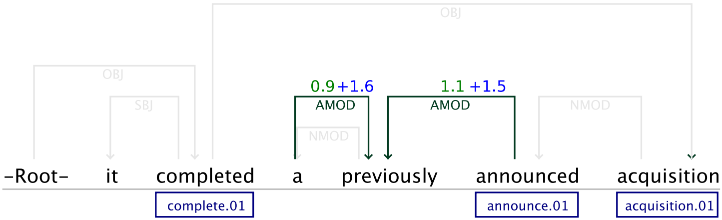
An incorrect competing dependency

What if the semantic score is not so dependant on syntax?



An incorrect compenting dependency with its syntactic score

What if the semantic score is not so dependant on syntax?



An incorrect compenting dependency with its score increased by the semantic score: hurt ↓ syntax

Discussion

- Why an incorrect syntactic dependency could have a high semantic score?
    - The semantic score is almost **independent** of the correct syntactic dependency.

Discussion

The semantic score is almost **independent** of the correct syntactic dependency:
it mainly relies on features extracted from the *modifier-predicate*

$$\text{semantic\_score}\left(h, m, l_{sem\ p_i}, x, \mathbf{w}, y'\right) =$$

$$\phi_{sem}\left(h, m, p_i, x, y'\right) \cdot \mathbf{w}^{(l_{sem\ p_i})}$$

Features are extracted by ϕ_{sem} from:

- h head
- m modifier
- p_i predicate
- h, m modifier-head
- m, p_i **modifier-predicate**
- h, p_i head-predicate

Posteval results

Group	Name	WSJ + Brown	WSJ	Brown
Lund (*)	Johansson (*)	85.49	86.61	76.34
Yahoo! (*)	Ciaramita (*)	82.69	83.83	73.51
HIT-IR	Che	82.66	83.78	73.57
Hong Kong (*)	Zhao (*)	82.24	83.41	72.70
Geneva (*)	Henderson (*)	80.48	81.53	71.93
Koc	Yuret	79.84	80.97	70.55
GSLT ML2	Samuelsson	79.79	80.92	70.49
DFKI 2	Zhang	79.32	80.41	70.48
NAIST	Watanabe	79.10	80.3	69.29
Antwerp	Morante	78.43	79.52	69.55
HIT-ICR	Li	78.35	79.38	70.01
UPC (*)	Lluís (*)	78.11	79.16	69.84
UT Austin	Baldridge	77.49	78.57	68.53
Koc	Yatbaz	77.45	78.43	69.61
USTC	Chen	77.00	77.95	69.23
Korea	Lee	76.90	77.96	68.34
Peking	Sun	76.28	77.1	69.58
Colorado	Choi	71.23	72.22	63.44
UAIC	Trandabat	63.45	64.21	57.41
DFKI 1	Neumann	19.93	20.13	18.14

Reasonable results for a built from scratch system.

It is one of the most efficient systems.

Future and Ongoing Work

1. Higher degree of joint processing
   - Joint predicate identification
   - No previous dependency parsing
2. Higher order dependencies
3. Improvement of the semantic classifier component
4. Projectivization techniques
5. Feature engineering and system tuning
6. Alternative joint models

Ongoing Work

Jointparser demo

http://www.lsi.upc.edu/~xlluis/jointparser

The end

Thank you

For further reading

- Xavier Carreras, Mihai Surdeanu and Lluís Màrquez
  Projective dependency parsing with perceptron.
  Proceedings of the CoNLL-2006, 2006.
- Ryan McDonald, Koby Crammer and Fernando Pereira
  Online large-margin training of dependency parsers.
  Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, 2005.
- James Henderson, Paola Merlo, Gabriele Musillo and Ivan Titov
  A Latent Variable Model of Synchronous Parsing for Syntactic and Semantic Dependencies.
  Proceedings of the CoNLL-2008, 2008.

For further reading

- Mihai Surdeanu, Lluís Màrquez, Xavier Carreras and Pere R. Comas
  Combination strategies for semantic role labeling.
  Journal of Artificial Intelligence Research, 2007.
- Nianwen Xue and Martha Palmer
  Calibrating features for semantic role labeling.
  Proceedings of the Empirical Methods in Natural Language Processing, 2004.
- Xavier Lluís and Lluís Màrquez
  A Joint Model for Parsing Syntactic and Semantic Dependencies.
  Proceedings of the CoNLL-2008, 2008.