High-performance Multilingual Semantic Role Labeling

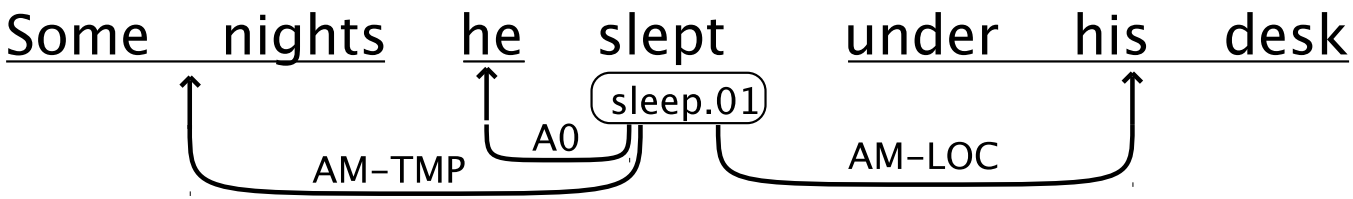
Anders Björkelund & Love Hafdell

May 26, 2009

Introduction

- Capture events and participants in text,

  who? what? where? when?



Semantic Roles

- Invariant under paraphrasing (as opposed to syntax), e.g.

  He slept under his desk some nights

  Some nights he was sleeping under his desk

- SRL is not an end-user application
- Intermediate step towards solving other problems
  - Information extraction
  - Document categorization
  - Automatic machine translation

Outline

- Background
- Constructed System
- Results
- Conclusion

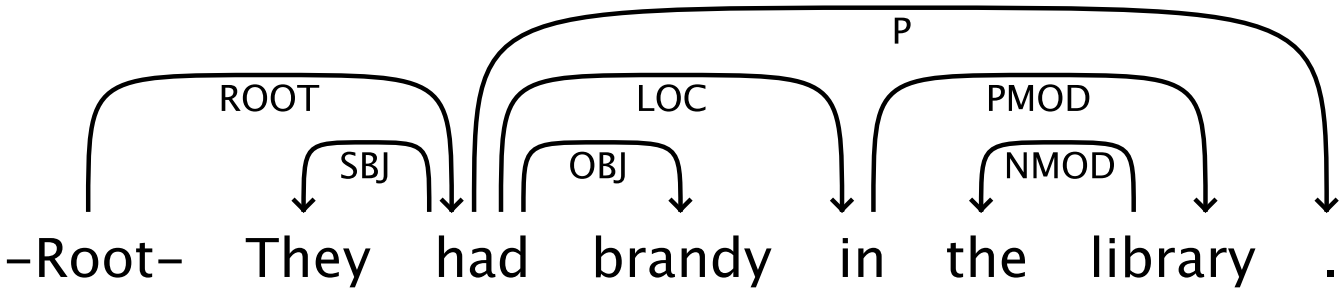
Corpus

- Corpus – vast "real world" text sample
- Annotation – meta-data such as
  - Lemma
  - Part of speech (POS)
  - Case, number, voice, etc (Feats)

	They	had	brandy	in	the	library
Lemma	they	have	brandy	in	the	library
POS	PRP	VBD	NN	IN	DT	NN

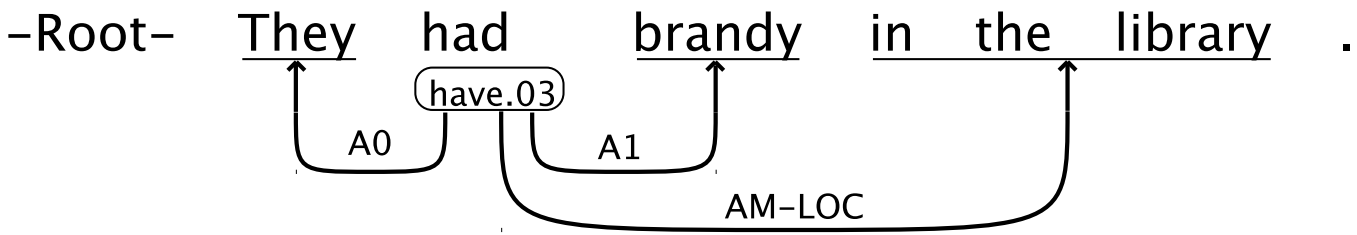
Syntactic Dependencies

- Binary head-dependent relations
- Formally a rooted directed acyclic graph (DAG)
- Labeled edges
- Artificial root node



Semantic Dependencies

- Events are denoted by *predicates*
- Predicates define a set of participants, *roles*
- Participants and adjuncts are called *arguments*



Semantic Frames

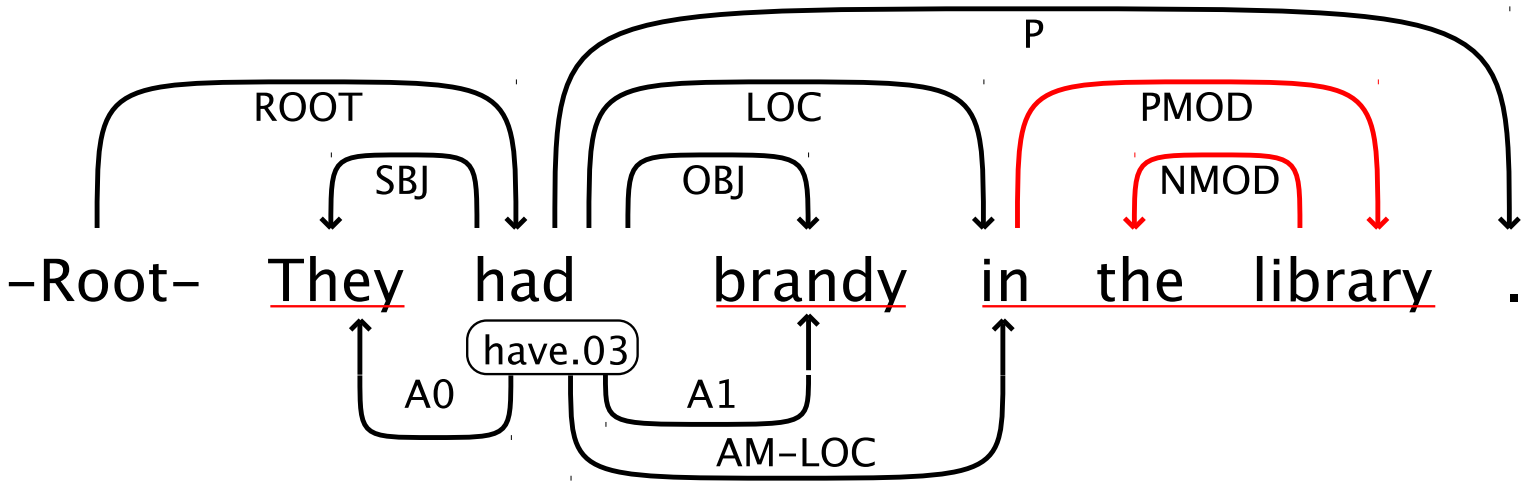
- Frames are defined by a lexicon
- Example from PropBank

```
<roleset id="have.03" vncls="-" name="own, possess">
<roles>
  <role n="0"  descr="owner"/>
  <role n="1"  descr="possession"/>
</roles>
</roleset>
```

- Specific lexicon for each language

Semantics and Dependency Grammar

- Semantic dependencies are also binary
- The *yield* of the argument node specifies the argument phrase.

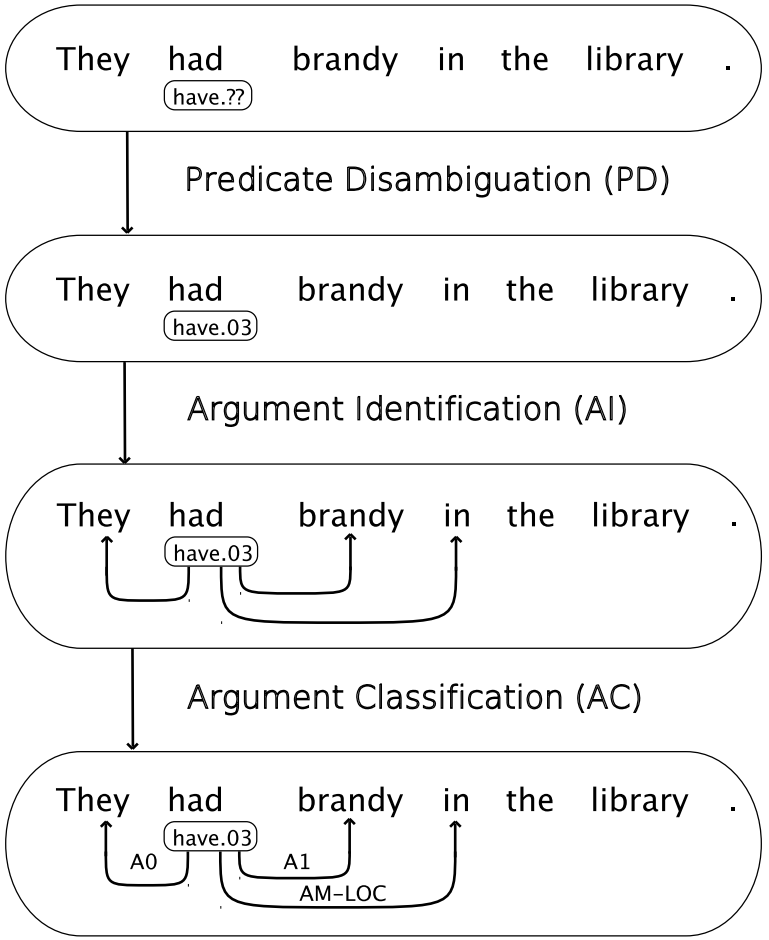


Machine Learning Tools

- LibLinear library used to train *classifiers*
- Feature representation $x_i \in X,\ X \subseteq \mathbb{R}^n$
- Class $y_i \in Y,\ Y = \{-1, +1\}$
- Training instances (x_i, y_i)
- The classifier is used to predict y given x
- Supports multi-class classification, i.e. $|Y| > 2$
- Logistic regression, outputs probabilities

The Baseline System

- Pipeline of classifiers
  - Predicate Disambiguation
  - Argument Identification
  - Argument Classification
- Requires annotated input
  - Lemma
  - Part of speech
  - Syntactic dependencies
  - Semantic dependencies (training only)
- Language-independent



Predicate Disambiguation (PD)

- Predicate frames grouped by lemma
- One classifier for each lemma

	They	had	brandy	in	the	library	.
have.03	0.852						
have.04	0.108						
have.02	0.0230						
have.01	0.0170						

Probability for all frames for the predicate have

Argument Identification (AI)

- Binary classifier considering each word in a sentence
- Generates an unlabeled proposition

	They	had	brandy	in	the	library	.
P(Arg)	0.979	0.00087	0.950	0.861	0.00006	0.0076	0.00009
P(¬Arg)	0.021	0.999	0.050	0.139	0.999	0.992	0.999

Probability that each word is an argument of had

Argument Classification (AC)

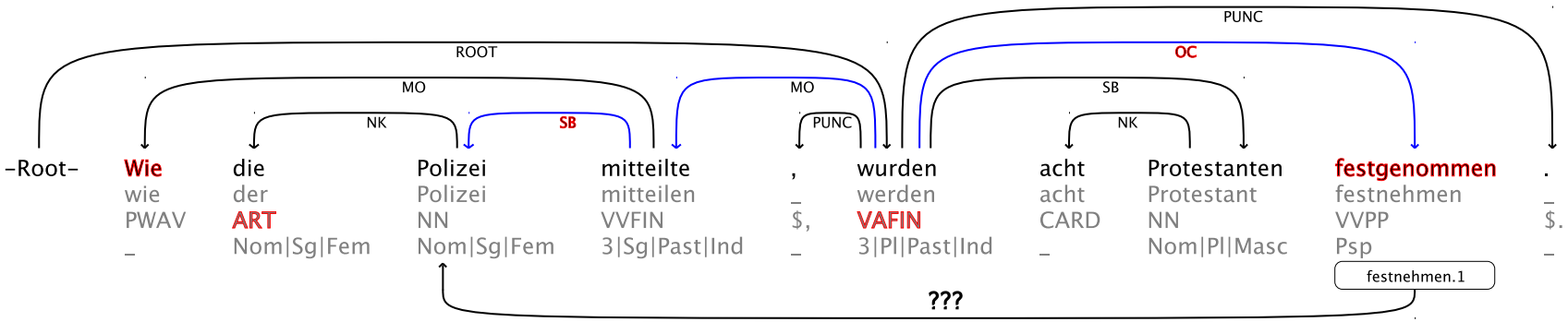
- Multiclass classifier, one class for each label

They		had	brandy		in		the	library	.
A0	0.999	-	A1	0.993	AM-TMP	0.471	-	-	-
A1	0.000487	-	C-A1	0.00362	AM-LOC	0.420	-	-	-
AM-DIS	0.000126	-	AM-ADV	0.000796	AM-MNR	0.0484	-	-	-
AM-ADV	0.000101	-	A0	0.000722	C-A1	0.00423	-	-	-

Probability of top four labels from the AC module for each argument of had

Features Used

- Predicate features,
  e.g. *PredWord, PredDeprel, PredParentPOS, etc*
- Argument features,
  e.g. *ArgDeprel, LeftPOS, LeftSiblingWord, etc*
- Path features, e.g. *DeprelPath, POSPath*



Feature Selection

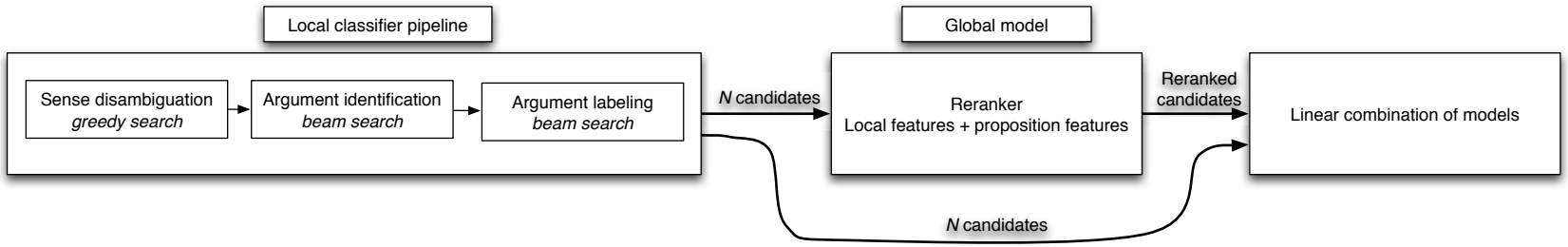
- Pool of 32 features
- Automatic procedure executed
  - For each step (i.e. PD, AI, AC)
  - For each language
- Greedy forward selection
- Incrementally adds the best feature

Shortcomings of the Pipeline

- Steps are executed sequentially
  - Error propagation
- Arguments are considered independently
  - Fails to catch the whole predicate argument structure

Reranker Extension

- Generation of N candidate propositions
- Reranker scores each candidate
- Pipeline and reranker are combined for final choice



Generation of Candidates (AI)

- AI module generates the top k unlabeled propositions

	They	had	brandy	in	the	library	.
P(Arg)	0.979	0.00087	0.950	0.861	0.00006	0.0076	0.00009
P(¬Arg)	0.021	0.999	0.050	0.139	0.999	0.992	0.999

- P_{AI} := the product of the probabilities of all choices

Example

- Using $k = 4$, we get the following unlabeled propositions

Proposition	P_{AI}
[They] had [brandy] [in] the library.	0.792
[They] had [brandy] in the library.	0.128
[They] had brandy [in] the library.	0.0417
They had [brandy] [in] the library.	0.0170

Generation of Candidates (AC)

- AC module generates the top *l* labellings of each proposition

They		had	brandy		in		the	library	.
A0	0.999	-	A1	0.993	AM-TMP	0.471	-	-	-
A1	0.000487	-	C-A1	0.00362	AM-LOC	0.420	-	-	-
AM-DIS	0.000126	-	AM-ADV	0.000796	AM-MNR	0.0484	-	-	-
AM-ADV	0.000101	-	A0	0.000722	C-A1	0.00423	-	-	-

- P_{AC} := the product of the probabilities of all labels

Example

- Using $l = 4$ and the most likely unlabeled proposition from last step, we get

Proposition	P_{AC}
[They]$_{\text{A0}}$ had [brandy]$_{\text{A1}}$ [in]$_{\text{AM-TMP}}$ the library.	0.467
[They]$_{\text{A0}}$ had [brandy]$_{\text{A1}}$ [in]$_{\text{AM-LOC}}$ the library.	0.417
[They]$_{\text{A0}}$ had [brandy]$_{\text{A1}}$ [in]$_{\text{AM-MNR}}$ the library.	0.0480
[They]$_{\text{A0}}$ had [brandy]$_{\text{A1}}$ [in]$_{\text{C-A1}}$ the library.	0.0421

Generation of Candidates

- AC probabilities are normalized by taking the geometric mean

$$P'_{AC} := (P_{AC})^{(1/a)}$$

- The probability of a labeled proposition is defined as

$$P_{Local} := P_{AI} \times P'_{AC}$$

- Pipeline probabilities are normalized to sum to 1

$$P'_{Local} := \frac{P_{Local}}{\sum P_{Local}}$$

The Reranker

- Binary classifier that considers complete propositions
- Features
  - AI features
  - AC features
  - Complete proposition features
- The reranker outputs a probability, $P_{Reranker}$
- Final candidate is selected to maximize $P_{Final} := P'_{Local} \times P_{Reranker}$

Selecting Final Candidate

Proposition	P'_{Local}	$P_{Reranker}$	P_{Final}
[They]$_{\text{A0}}$ had [brandy]$_{\text{A1}}$ [in]$_{\text{AM-TMP}}$ the library.	0.306	0.359	0.110
[They]$_{\text{A0}}$ had [brandy]$_{\text{A1}}$ [in]$_{\text{AM-LOC}}$ the library.	0.295	0.246	0.0726
[They]$_{\text{A0}}$ had [brandy]$_{\text{A1}}$ in the library.	0.0636	0.451	0.0287
[They]$_{\text{A0}}$ had [brandy]$_{\text{A1}}$ [in]$_{\text{AM-MNR}}$ the library.	0.143	0.0890	0.0128
[They]$_{\text{A0}}$ had [brandy]$_{\text{A1}}$ [in]$_{\text{C-A1}}$ the library.	0.137	0.0622	0.00854
[They]$_{\text{A0}}$ had brandy [in]$_{\text{AM-TMP}}$ the library.	0.0139	0.0206	$2.86 \cdot 10^{-4}$
[They]$_{\text{A0}}$ had brandy [in]$_{\text{AM-LOC}}$ the library.	0.0131	0.0121	$1.58 \cdot 10^{-4}$
They had [brandy]$_{\text{A1}}$ [in]$_{\text{AM-TMP}}$ the library.	0.00452	0.0226	$1.02 \cdot 10^{-4}$
They had [brandy]$_{\text{A1}}$ [in]$_{\text{AM-LOC}}$ the library.	0.00427	0.0133	$5.68 \cdot 10^{-5}$
[They]$_{\text{A0}}$ had brandy [in]$_{\text{AM-MNR}}$ the library.	0.00445	0.00364	$1.62 \cdot 10^{-5}$

Top ten propositions sorted by final score

Evaluation Measures

- Labeled attachment score (LAS) – Syntax accuracy (input)
- Semantic F_1 ($SemF_1$) – Overall system accuracy
- Predicate F_1 ($PredF_1$) – PD accuracy
- Argument F_1 ($ArgF_1$) – AI and AC accuracy
- Scores range from 0 (bad) to 100 (good)

Baseline Results

- Evaluation figures on test sets

	LAS	PredF_1	ArgF_1	SemF_1
Catalan	86.13	87.20	76.13	79.54
Chinese	78.46	94.92	69.83	77.84
Czech	78.46	94.20	74.24	84.99
English	85.50	95.59	79.29	84.44
German	85.93	81.45	77.44	79.01
Japanese	91.12	99.07	60.26	75.61
Spanish	86.20	85.43	76.56	79.28
Average	84.54	91.12	73.39	80.10

Reranker Results

- Results and improvement by reranker ($SemF_1$ scores)

	Baseline	Reranker	Gain
Catalan	79.54	80.01	0.47
Chinese	77.84	78.60	0.76
Czech	84.99	85.41	0.42
English	84.44	85.63	1.19
German	79.01	79.71	0.70
Japanese	75.61	76.30	0.69
Spanish	79.28	79.91	0.63
Average	80.10	80.80	0.70

Contribution to CoNLL Shared Task

- Results in CoNLL 2009 Shared Task (SemF$_1$ scores)

	Team	Average	Catalan	Chinese	Czech	English	German	Japanese	Spanish
1	Hong Kong	80.47	**80.32**	77.72	85.19	85.44	75.99	78.15	**80.46**
2	Lund†	80.31	80.01	**78.60**	85.41	85.63	**79.71**	76.30	76.52
3	Hong Kong	79.96	80.10	76.77	82.04	**86.15**	76.19	78.17	80.29
4	Harbin	79.94	77.10	77.15	**86.51**	85.51	78.61	**78.26**	76.47
5	Geneva	78.42	77.44	76.05	86.02	83.24	71.78	77.23	77.19
6	Edinburgh	77.46	78.00	77.73	75.75	83.34	73.52	76.00	77.91
7	Berkeley	76.00	74.53	75.29	79.02	80.39	75.72	72.76	74.31
8	NIST	75.65	72.35	74.17	84.69	84.26	63.66	77.93	72.50
9	Brown⋄	72.85	72.18	72.43	78.02	80.43	73.40	61.57	71.95
10	Hefei	70.78	66.34	71.57	75.50	78.93	67.43	71.02	64.64
11	DFKI	70.31	67.34	73.20	78.28	77.85	62.95	64.71	67.81
12	Harbin	69.72	66.95	67.06	79.08	77.17	61.98	69.58	66.23
...									

Summary

- Dependency-based automatic SRL system
- Pipeline of linear classifiers
- Further extended with reranker
- State-of-the-art performance in seven languages
- Room for improvements

Reranker Potential

- Upper bound on reranker

	Baseline	Reranker	Upper Bound
Catalan	79.54	80.01	90.11
Chinese	77.84	78.60	88.70
Czech	84.99	85.41	92.55
English	84.44	85.63	93.80
German	79.01	79.71	88.78
Japanese	75.61	76.30	90.04
Spanish	79.28	79.91	89.12
Average	80.10	80.80	90.44

Improvements

- Reranker features
  - Feature pool
  - Feature selection
- Review combination of pipeline and reranker
- Dynamic beam width

The End

Thank you.