A Comparison of Structural Correspondence Learning and Self-training for Discriminative Parse Selection

Barbara Plank
b.plank@rug.nl

University of Groningen (RUG)
The Netherlands

NAACL HLT 2009 Workshop on
Semi-supervised Learning for Natural Language Processing

June 4, 2009

The Problem: Domain dependence

- Train a model on data you have; test it, works pretty good
- However, whenever **test** and **training data** differ, the performance of such a supervised system degrades considerably (Gildea, 2001)

The Problem: Domain dependence

- Train a model on data you have; test it, works pretty good
- However, whenever **test** and **training data** differ, the performance of such a supervised system degrades considerably (Gildea, 2001)

Possible solutions:

1. Build a model for every domain we encounter → Expensive!
2. Adapt a model from a *source* domain to a *target* domain
   → **Domain Adaptation**

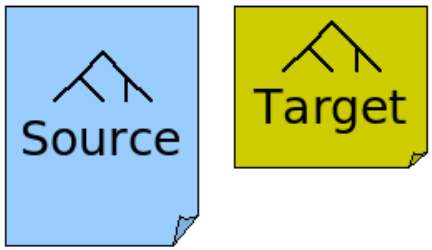
Approaches to Domain Adaptation

Recently gained attention - Approaches (Daumé III, 2007):

Approaches to Domain Adaptation

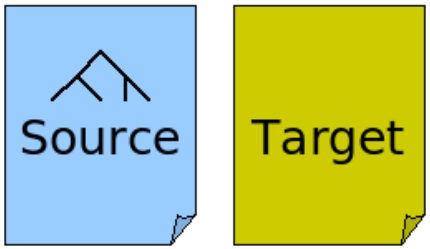
Recently gained attention - Approaches (Daumé III, 2007):

a. Supervised Domain Adaptation



- Limited annotated resources in new domain (Gildea, 2001; Chelba and Acero, 2004; Hara, 2005; Daumé III, 2007)

b. Semi-supervised Domain Adaptation



- No annotated resources in new domain (Blitzer et al., 2006; McClosky et al., 2006; McClosky and Charniak, 2008) – more difficult, but also more realistic scenario

Semi-supervised Adaptation for Parse Selection

Motivation

- Adaptation of Parse Selection Models - less studied area
- Most previous work on parser adaptation for data-driven systems
  - Data-driven systems (e.g. PCFGs) - (usually) one-stage
  - Two-stage: Hand-crafted grammar with separate disambiguation
- Few studies on adapting disambiguation models (Hara, 2005; Plank and van Noord, 2008) focused exclusively on the supervised case

Semi-supervised Adaptation for Parse Selection

Motivation

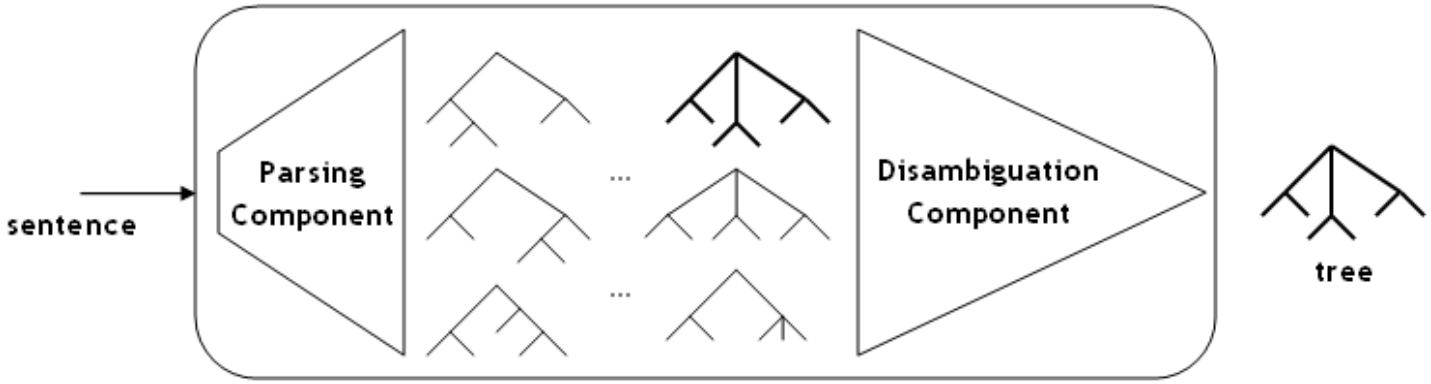
- Adaptation of Parse Selection Models - less studied area
- Most previous work on parser adaptation for data-driven systems
  - Data-driven systems (e.g. PCFGs) - (usually) one-stage
  - Two-stage: Hand-crafted grammar with separate disambiguation
- Few studies on adapting disambiguation models (Hara, 2005; Plank and van Noord, 2008) focused exclusively on the supervised case

Semi-supervised Adaptation: How can we exploit unlabeled data?

1. Structural Correspondence Learning (SCL)
   - A recent attempt at EACL-SRW 2009 (Plank, 2009) shows promising results of SCL for Parse Selection
2. Self-training
   - What do we reach with self-training?

Background: Alpino Parser

- Two-stage dependency parser for Dutch



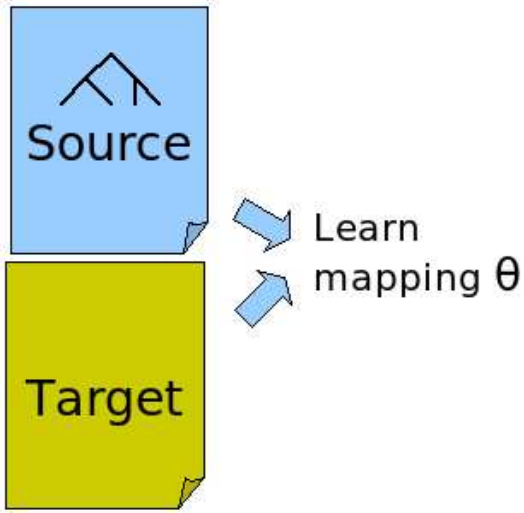
- HPSG-style grammar rules, large hand-crafted lexicon
- Conditional Maximum Entropy Disambiguation Model:
  - Feature functions f_j / weights w_j
  - Estimation based on *Informative samples* (Osborne, 2000)

$$p_\theta(\omega|s;w) = \frac{1}{Z_\theta} q_0 \exp\left(\sum_{j=1}^{m} w_j f_j(\omega)\right)$$

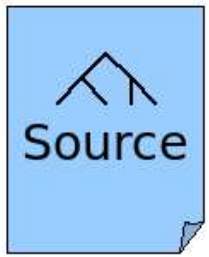
- Output: Dependency Structure

Structural Correspondence Learning (SCL) - Idea

- Domain adaptation algorithm for feature based classifiers, proposed by Blitzer et al. (2006), based on Ando and Zhang (2005)
- Use data from both source and target domain to induce correspondences among features from different domains
- Incorporate correspondences as new features in the labeled data of the source domain



Learn w,v on source data:



$x \cdot w + (x \cdot \theta) \cdot v$

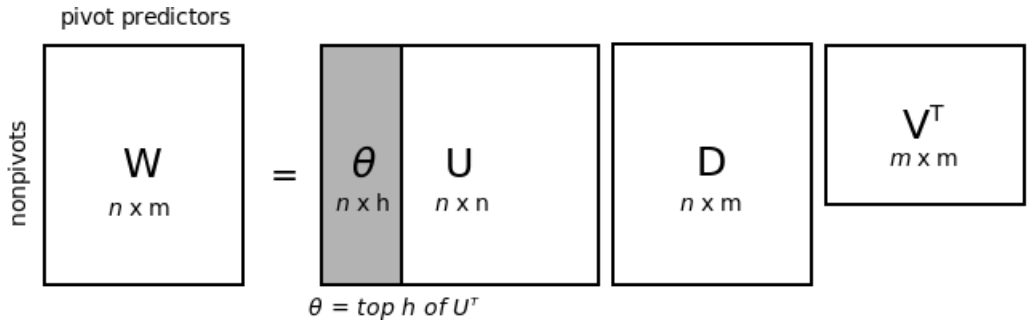
Structural Correspondence Learning (SCL) - Idea

Find correspondences through pivot features:

$$\underset{\text{domain } A}{\textit{feat}_X} \leftrightarrow \underset{(\text{“linking” feature})}{\text{pivot feature}} \leftrightarrow \underset{\text{domain } B}{\textit{feat}_Y}$$

SCL - Algorithm:

1. Select pivot features.
2. Train a binary classifier for every pivot features.
3. Dimensionality Reduction. Arrange pivot predictor weight vectors in matrix W. Apply SVD to W, and select the h top left singular vectors θ.
4. Train a new model on the source data augmented with $x \cdot \theta$.



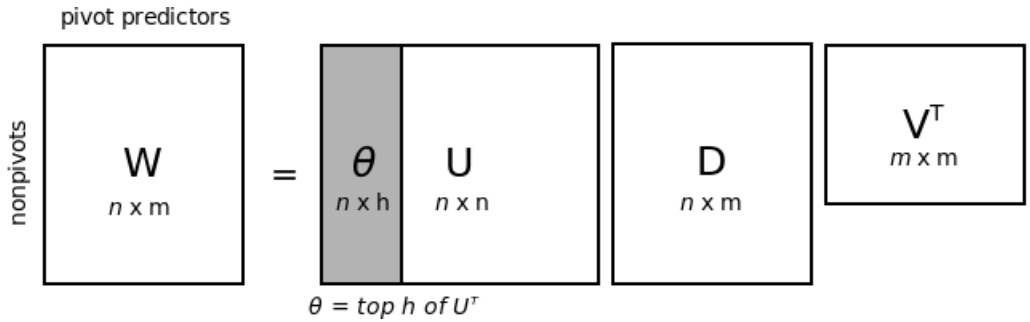
Structural Correspondence Learning (SCL) - Idea

Find correspondences through pivot features:

$$\underset{\textit{domain A}}{\textit{feat}_X} \leftrightarrow \underset{(\text{“linking” feature})}{\text{pivot feature}} \leftrightarrow \underset{\textit{domain B}}{\textit{feat}_Y}$$

SCL - Our instantiation:

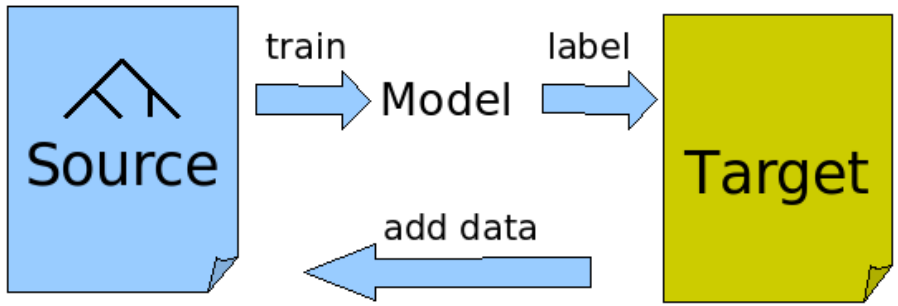
1. **Parse** unlabeled data $\rightarrow$ **Features**: properties of parses
2. Select pivot features. **Our Pivots**: frequent grammar rules (mainly)
3. Train a binary classifier for every pivot features.
4. Dimensionality Reduction. Arrange pivot predictor weight vectors in matrix W. Apply SVD to W, and select the h top left singular vectors θ.
5. Train a new model on the source data augmented with $x \cdot \theta$.



Self-training

What is Self-training?

- A general semi-supervised bootstrapping algorithm
- Procedure: An existing model labels unlabeled data. The newly labeled data is then taken at face value and combined with the actual labeled data to train a new model. This process can be iterated.



Self-training

We examine several self-training variants:

- Multiple versus single iteration
- Selection versus no selection (taking all self-labeled data or not)
- Delibility versus indeliblity for multiple iterations (Abney, 2007)

Notion of (in)delibility (Abney, 2007):

- *delible case*: classifier relabels all of unlabeled data from scratch in every iteration; it may become unconfident about previous labeled instances and they may drop out
- *indelible case:* labels once assigned do not change

Self-training: Previous work

- Most studies focused data driven systems (Steedman et al., 2003; McClosky et al., 2006; Reichart and Rappoport, 2007; McClosky and Charniak, 2008; McClosky et al., 2008)

	Parser type	Seed size	Iterations	Improved?
Charniak (1997)	Generative	Large	Single	
McClosky et al. (2006)	Gen.+Disc.	Large	Single	
Steedman et al. (2003)	Generative	Small	Multiple	
Reichart & Rappoport (2007)	Generative	Small	Single	

Table: Summary of self-training for parsing (table from McClosky et al., 2008)

(large = 40k sents, small = $<$ 1k sents)

Self-training: Previous work

- Most studies focused data driven systems (Steedman et al., 2003; McClosky et al., 2006; Reichart and Rappoport, 2007; McClosky and Charniak, 2008; McClosky et al., 2008) – different results

	Parser type	Seed size	Iterations	Improved?
Charniak (1997)	Generative	Large	Single	**No**
McClosky et al. (2006)	Gen.+Disc.	Large	Single	**Yes**
Steedman et al. (2003)	Generative	Small	Multiple	**No**
Reichart & Rappoport (2007)	Generative	Small	Single	**Yes**

Table: Summary of self-training for parsing (table from McClosky et al., 2008)

(large = 40k sents, small = $<$ 1k sents)

- How good is self-training for discriminative parse selection?

Experimental design

Data

- General, out-of-domain: Alpino (newspaper; 7k sents/145k tokens)
- Domain-specific: Wikipedia articles

Construction of target data from Wikipedia (WikiXML)

- Exploit Wikipedia's category system (XQuery,Xpath): extract pages related to p (through sharing a direct, sub- or super category)
- Overview of collected unlabeled target data:

Dataset	Size	Relationship
Prince	290 articles, 145k tokens	filtered super
Pope Johannes Paulus II	445 articles, 134k tokens	all
De Morgan	394 articles, 133k tokens	all

Evaluation metric: Concept Accuracy (labeled dependency accuracy)

Experiments & Results

	Accuracy	E.R.
baseline Prince	85.03	-
Oracle	88.70	-
SCL	$\star$ 85.30	7.34
baseline Paus	85.72	-
Oracle	89.09	-
SCL	85.82	2.81
baseline DeMorgan	80.09	-
Oracle	83.52	-
SCL	80.15	1.88

Table: Result of SCL and Self-training (accuracy and error reduction). Entries marked with $\star$ are significant at $p < 0.05$).

- SCL: small but consistent increase in accuracy

Experiments & Results

	Accuracy	E.R.
baseline Prince	85.03	-
Oracle	88.70	-
SCL	$\star$ 85.30	7.34
Self-train (all)	85.08	1.46
baseline Paus	85.72	-
Oracle	89.09	-
SCL	85.82	2.81
Self-train (all)	85.78	1.71
baseline DeMorgan	80.09	-
Oracle	83.52	-
SCL	80.15	1.88
Self-train (all)	80.24	4.65

Table: Result of SCL and Self-training (accuracy and error reduction). Entries marked with $\star$ are significant at $p < 0.05$).

- SCL: small but consistent increase in accuracy
- Self-training (all at once, no selection, single iteration): roughly baseline accuracy (exception on DeMorgan dataset)
- Work in progress

Experiments & Results

	Accuracy	E.R.
baseline Prince	85.03	-
Oracle	88.70	-
SCL	$\star$ 85.30	7.34
Self-train (all)	85.08	1.46
baseline Paus	85.72	-
Oracle	89.09	-
SCL	85.82	2.81
Self-train (all)	85.78	1.71
baseline DeMorgan	80.09	-
Oracle	83.52	-
SCL	80.15	1.88
Self-train (all)	80.24	4.65

Table: Result of SCL and Self-training (accuracy and error reduction). Entries marked with $\star$ are significant at $p < 0.05$).

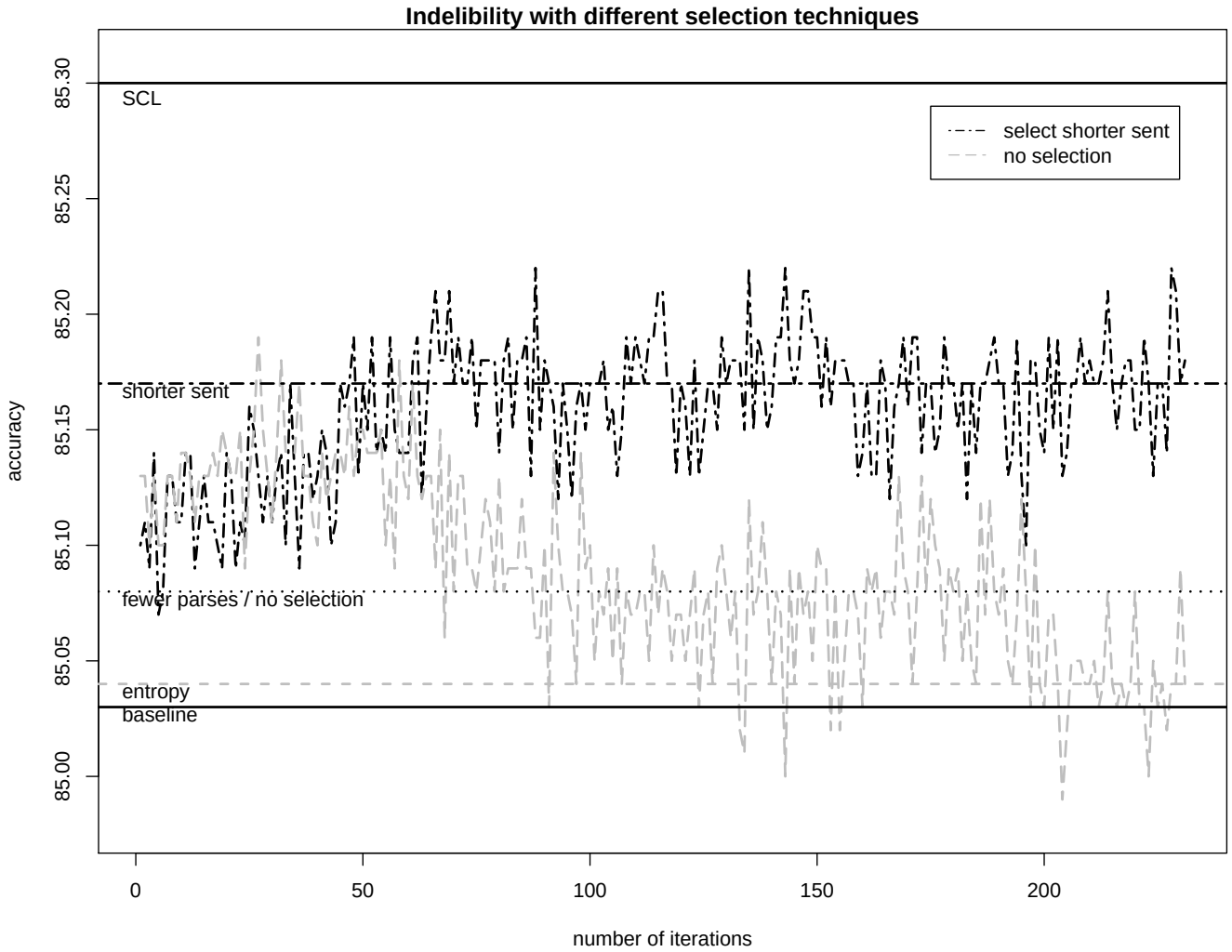
- SCL: small but consistent increase in accuracy
- Self-training (all at once, no selection, single iteration): roughly baseline accuracy (exception on DeMorgan dataset)
- Work in progress
- Are other instantiations of self-training more effective?

Experimental design

Self-training

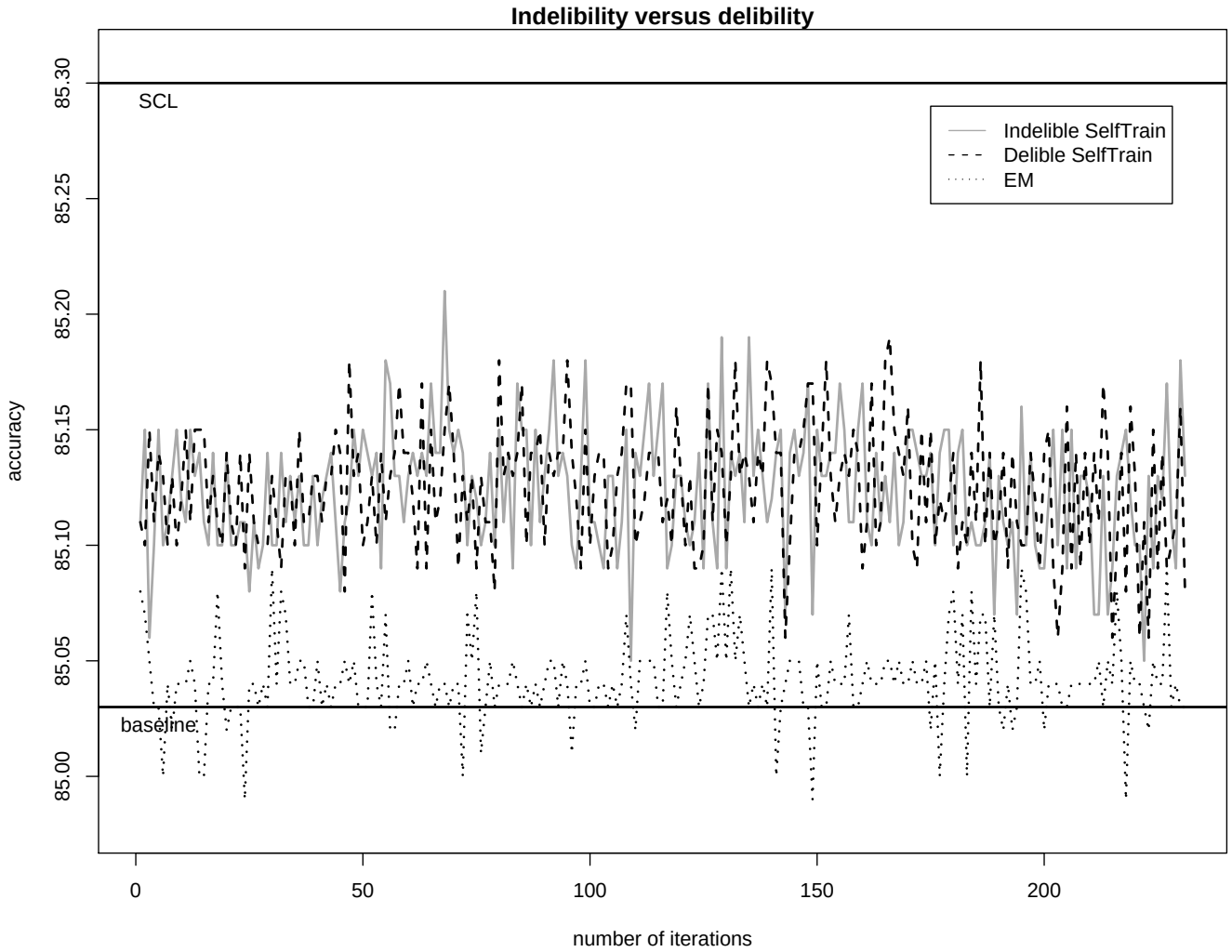
- For the iterative setting, we follow Steedman et al. (2003): Parse 30 sentences from which 20 are selected in every iteration
- Scoring methods:
  - Entropy: $-\sum_{y \in Y(s)} p(\omega|s,\theta) \log p(\omega|s,\theta)$
  - Number of parses: $|Y(s)|$
  - Sentence Length: $|s|$

Self-training results



- Selection vs. no selection: no selection degrades performance
- Running multiple is on average just the same as running a single iteration

Self-training results



Delible versus indelible self-training achives *very* similar performance → indelibility preferred (much faster)

Conclusions

- Examination of SCL and self-training for Parse Selection on Wikipedia domains
- SCL slightly but constantly outperformed the baseline
- Self-training achieves roughly baseline performance; none of the evaluated variants achieves a significant improvement over the baseline
- The preliminary evaluation favors the use of SCL over self-training, although the findings are not confirmed on all testsets
- Applying SCL involves many design choices and practical issues
- Future work
  a. Further explore/refine SCL (other testsets, varying amount of target domain data, pivot selection, etc.)
  b. Other ways to exploit unlabeled data (e.g. more 'direct' mapping between features?)

Thank you for your attention.

Wikipedia article	Accuracy	base	oracle	sent
Prince (musician)	85.03	71.95	88.70	357
Paus Johannes Paulus II	85.72	74.30	89.09	232
Augustus De Morgan	80.09	70.08	83.52	254

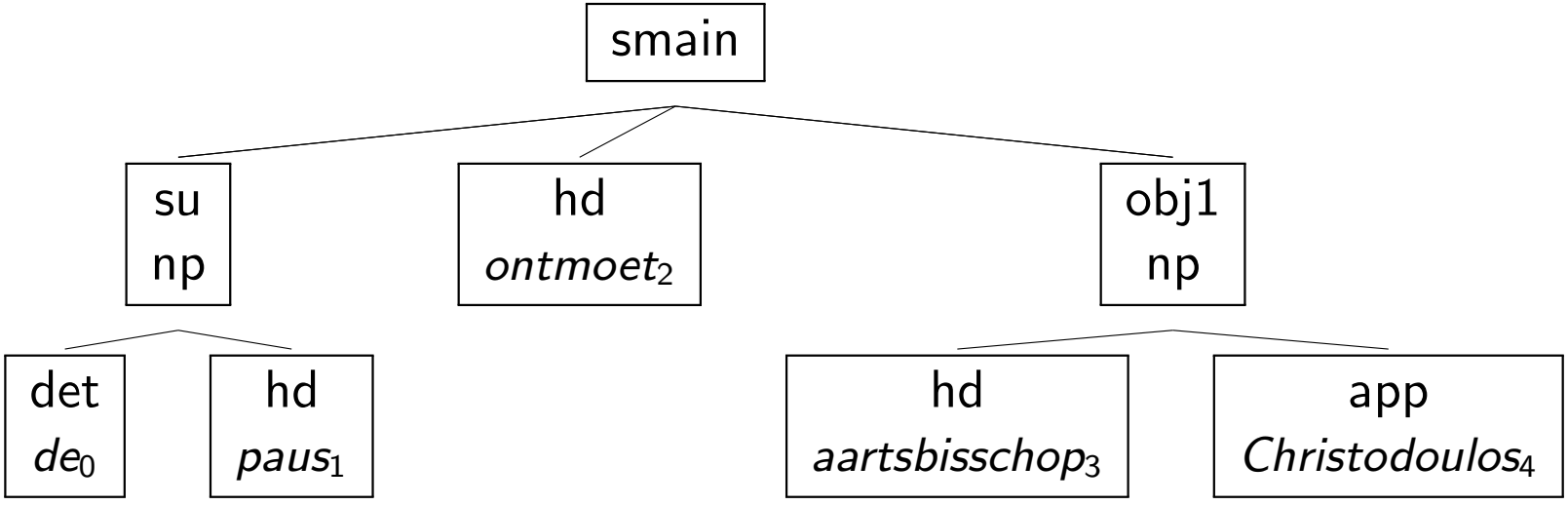
Table: Supervised Baseline results.

	CA	ϕ
Prince baseline	85.03	78.06
SCL	85.30	79.67
SVD, Dim=25	85.26	79.44
SVD, Dim=50	85.28	79.58
Paus baseline	85.72	77.23
SCL	85.82	77.87
SVD, Dim=25	85.70	77.10
SVD, Dim =50	85.72	77.23
DeMorgan baseline	80.09	74.44
SCL	80.15	74.92
SVD, Dim=25	80.15	74.92
SVD, Dim=50	80.22	75.42

Table: 'Basque SVD': variant of SCL, inspired by work of Agirre E. and Lopez de Lacalle O.

Parse and Features

Example: *De paus ontmoet aartsbisschop Christodoulos*
(The pope meets archbishop Christodoulos)




```
f1(noun)
f1(name(PER))
f1(verb(transitive))
f2(Christodoulos,name(PER))
f2(ontmoet,verb(transitive))
appos_person(PER,aartsbisschop)
```

```
r1(np_det_n)
r1(np_n)
dep23(noun,hd/su,verb)
dep23(name(PER),hd/app,noun)
dep34(aartsbisschop,noun,hd/obj1,verb)
dep34(paus,noun,hd/su,verb)
```

Appendix

Pivot features - Examples

```
r1(np_det_n)
r1(n_adj_n)
r1(n_n_adv)
r1(pron_pron_rel)
s1(subj_topic)
s1(non_long_distance_dep)
s1(non_subj_topic)
```