Newswire-to-law Adaptation of Graph-based Dependency Parsers

Barbara Plank and Anders Søgaard
barbara.plank@disi.unitn.it
soegaard@hum.ku.dk

University of Trento
University of Copenhagen

January 24, 2012
Evalita 2011, Rome

The Problem: Domain dependence

A very common problem/situation in NLP:

- 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)

WIKIPEDIA
The Free Encyclopedia

The Problem: Domain dependence

A very common problem/situation in NLP:

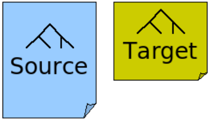
- Train a model on data you have; test it, works pretty good
- However, whenever **test** and **training data** differ, the performance of such a supervised system degrades considerably (Gildea, 2001)

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**

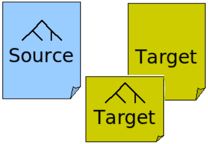
Approaches to Domain Adaptation (DA)

- Supervised Domain Adaptation



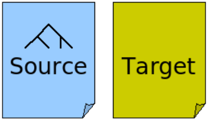
- Limited annotated resources in new domain (Gildea, 2001; Daumé III, 2007)

- Semi-supervised Domain Adaptation



- Less explored; started to gain attention only recently (Daumé III, 2010, MW Chang, 2010)

- Unsupervised Domain Adaptation[1]



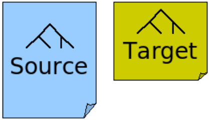
- No annotated resources in new domain (McClosky et al., 2006; Blitzer et al., 2006)

---

[1] Until 2010 often called semi-supervised DA (cf. Plank, 2011)

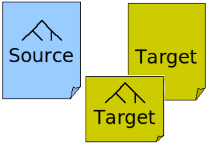
Approaches to Domain Adaptation (DA)

- Supervised Domain Adaptation



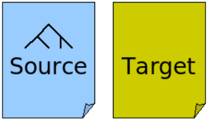
- Limited annotated resources in new domain (Gildea, 2001; Daumé III, 2007)

- Semi-supervised Domain Adaptation → Task 1



- Less explored; started to gain attention only recently (Daumé III, 2010, MW Chang, 2010)

- Unsupervised Domain Adaptation[1] → Task 2



- No annotated resources in new domain (McClosky et al., 2006; Blitzer et al., 2006)

[1] Until 2010 often called semi-supervised DA (cf. Plank, 2011)

Our participation in Evalita 2011

We participated in Task 2 (Unsupervised DA) - Question:

- How far can we go by exploiting only **unlabeled** data? Without any hand-correction?

Our participation in Evalita 2011

We participated in Task 2 (Unsupervised DA) - Question:

- How far can we go by exploiting only **unlabeled** data? Without any hand-correction?

Experimental Setup

- Base parser: MSTParser
  - Graph-based dependency parser (minimum spanning tree)
  - Not specific to Italian, needs CoNLL training data
  - Second-order projective parsing mode with 2-best MIRA
- **Source domain:** newspaper text (Italian ISST-TANL corpus)
  - Train: 70k tokens, 3,2k sents
- **Target domain:** legal text
  - Devel: 5k tokens, 147 sents
  - Unlabeled: 1,300k tokens, 620k sents

Results: Baseline

What's the parser's performance before adaptation?

Results: Baseline

What's the parser's performance before adaptation?

	LAS	UAS
source devel	78.59	83.87
target devel	76.45	80.67

Table: MSTParser out-of-the box (trained on source data); LAS: Labeled Attachment Score; UAS: Unlabeled Attachment Score.

Results: Baseline

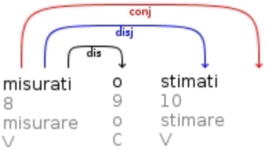
What's the parser's performance before adaptation?

	LAS	UAS
source devel	78.59	83.87
target devel	76.45	80.67

Table: MSTParser out-of-the box (trained on source data); LAS: Labeled Attachment Score; UAS: Unlabeled Attachment Score.

source devel (+nf)	80.19	86.00
target devel (+nf)	76.96	81.22

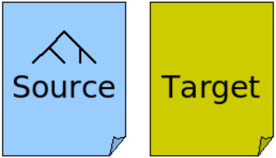
Table: MSTParser with one new feature (+nf) template (labels for siblings) since annotation distinguishes coordination types (conj/disj).



Adapting the parser to law text

Ways to use unlabeled data:

1. Exploiting unlabeled target data
   - 2 methods tested
2. Exploiting automatically labeled target data
   - Use base parser to annotate pool of unannotated data
   - 3 methods tested



Exploiting unlabeled target data (1/2)

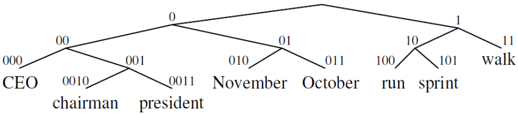
Instance weighting

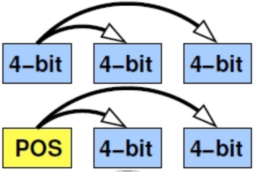
- Intuition: weigh each instance in the source data by the probability it was sampled from the target domain
- Implementation: weighting the loss function of the MIRA online algorithm
- Text classifier (trigrams) used to approximate probability distribution and obtain instance weights
- Technically: retrain MSTParser on source by including instance weights
- Result: **did not work**; far below baseline

Exploiting unlabeled target data (2/2)

Word clusters

- Intuition: address lexical sparsity by clustering words according to contextual similarity
- Implementation: Brown algorithm used to induce clusters from source and target data
- Technically: add new features to MSTParser that replace words with cluster indices; different bit-string prefixes give different granularity





Exploiting unlabeled target data (2/2)

Word clusters

- Examples:

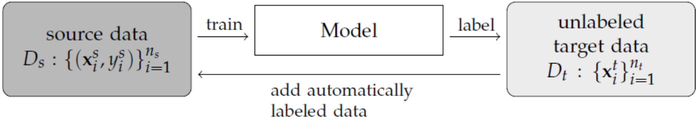
0000100	articolo	100101100	dare
0000100	art	100101100	prendere
0000101	art.	100101101	avere
00001100	paragrafo	1001011101	revocare
00001100	comma	1001011101	incaricare
000011010	paragrafi	1001011101	nominare

- Result: **did not work either**; better than instance weighting but still below baseline (LAS 71%)
- Too many new features? Bad clusters?

Exploiting automatically labeled target data (1/3)

Self-training

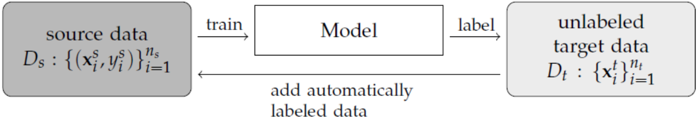
- Take auto-labeled data at face value and add to source



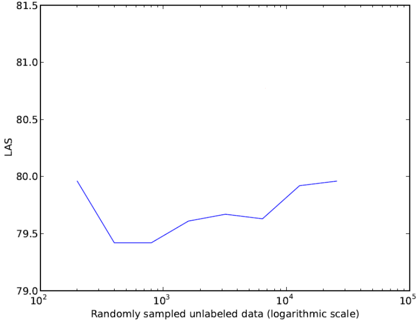
Exploiting automatically labeled target data (1/3)

Self-training

- Take auto-labeled data at face value and add to source



- Result: **did not work**



Exploiting automatically labeled target data (2/3)

Co-training

- Take auto-labeled data two parsers agree upon
- We used MSTParser and Bohnet's parser
- Results: improved over baseline, approximately +0.3% LAS (on 58k unique sentences the parsers agreed upon)

Exploiting automatically labeled target data (3/3)

Dependency triplets

- Extract named dependency relations $r(w_1, w_2)$ from auto-labeled target data $\rightarrow$ learn bilexical preferences
- Calculate normalized point-wise mutual information score:

$$npmi = (\log \frac{f(r(w_1, w_2))}{f(r(w_1, \_))f(r(\_, w_2))})/ - \log f(r(w_1, w_2))$$

- Example triplets:

```
0.726149069696 obj informare autorità
0.653647108129 obj adire autorità
0.628772868532 obj consultare autorità
0.9217 mod Stati membro
0.4594 mod Stati extracomintario
```

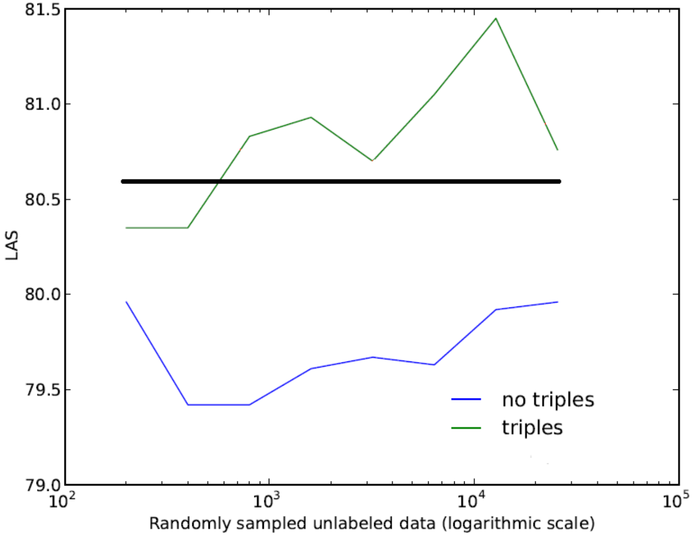
Dependency triplets

- Integrate triplets into parser as new features
- A new feature $z(t, r)$ for every major Pos tag t and relation r, e.g. for $obj(write, article)$ add new feature $z(VB, OBJ)$ with score given by NPMI (binned into buckets) $\rightarrow$ small amount of new features
- Results:

target devel	76.96 (79.53)	81.22	89.26
target devel with triplets	**78.19** (80.54)	82.57	90.23

Table: MSTParser with auto-parsed dependency triplets. Score in parenthesis is by excluding punctuation from scoring.

Self-training with and without triplets



Submitted Results

- Using only triplets
- Using selftraining with triplets (with 12k sentences added)
- Results on target test set:

	LAS	UAS
model before adaptation	74.62	78.22
self-training with triplets	74.30	78.05
triplets only	74.02	77.92

Table: Results on released test data

- Result: Just around baseline performance (slightly below)

Conclusions and Future Work

- Improvements observed on development data did not carry over to test set
- Why?
  - Overfitting base model on small amounts of training data?
  - Do we need hand-corrected data? However, adding target devel to source gives only limited improvement:

	LAS	UAS
model before adaptation	74.62	78.22
supervised (source+target dev)	75.95	79.47

Table: Results on released test data

  - Systematic errors in formal law texts? Properties of the data?

Conclusions and Future Work

- A first look - One pecularity: enumerations
- Parser got attachment of enumeration wrong in:
  `( 8 ) Il presente regolamento non dovrebbe ..`
  while it was often correct in similar cases such as:
  `a ) ‘‘ vettore aereo ’’`
- Influence of different POS tags? `8/N` vs. `a/S` (in PTB both would be LS, list item markers i.e. S?)

	LAS	UAS
model before adaptation (original data)	74.62	78.22
model before adaptation (changing POS)	76.47	80.52
model after adaptation (triplets)	77.17	81.37

Table: Results on released test data with `x/N` changed to `x/S` (24x)

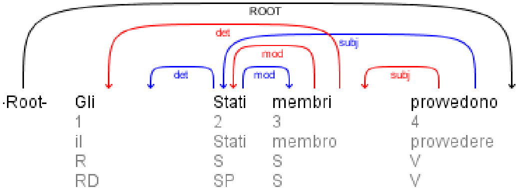
- Need deeper error analysis & larger evaluation set

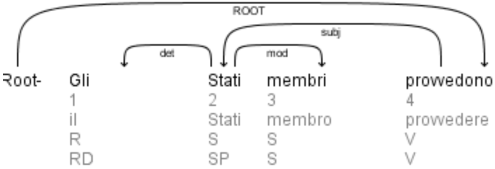
Questions? Comments? Suggestions?

Thank you.

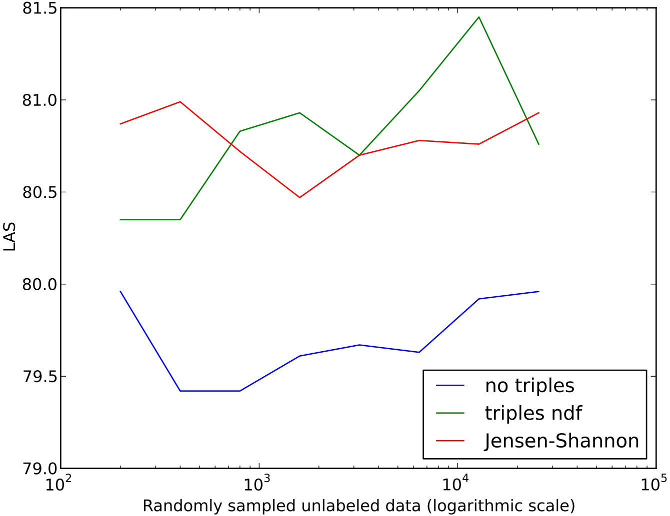
The first author would like to thank
iKernels (DISI) and PARLI for supporting this research.

Gli Stati membri





Self-training with and without triplets



Sentence Length vs LAS