Introduction and Motivation

Subdomain Sensitive Statistical Parsing
- Subdomain Sensitive Parsers
- Parser Combination Techniques

Experiments and Results

Conclusions and Future Work

Subdomain Sensitive Statistical Parsing using Raw Corpora

Barbara Plank[1] and Khalil Sima'an[2]

[1] Alfa Informatica, Faculty of Arts
University of Groningen, The Netherlands
b.plank@rug.nl

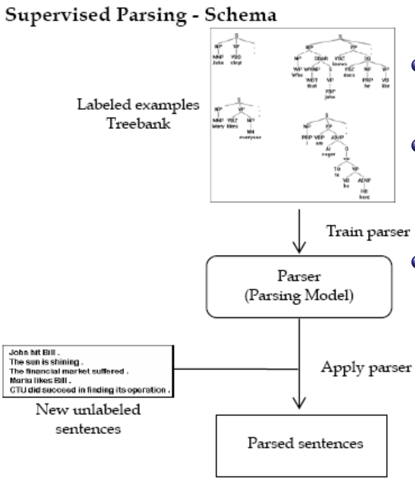
[2] Language and Computation, Faculty of Science
University of Amsterdam, The Netherlands
simaan@science.uva.nl

LREC 2008
Marrakech, Morocco

Outline

1. Introduction and Motivation
2. Subdomain Sensitive Statistical Parsing using Raw Corpora
   - Subdomain Sensitive Parsers
   - Parser Combination Techniques
3. Experiments and Results
4. Conclusions and Future Work

Statistical parsing



- Problem: Ambiguity of natural language sentences
- Common approach: Train a parser/model on a treebank. Apply to new input.
- Variations: phrase/dependency structure, formal grammar, statistical model and estimator.

Motivation

Is there more in a treebank that we might exploit?

- We view a treebank as a mixture of subdomains, each addressing certain concepts more than others

  "politics, stock market, financial news etc. can be found in the WSJ" (Kneser and Peters, 1997)

- The parsing statistics gathered from the treebank are averages over different subdomains,
- Averages smooth out the differences between subdomains and weaken the biases

1. Do subdomains matter?
2. How to incorporate subdomain sensitivity into an existing state-of-the-art parser?

Motivation - Our Approach

Subdomains $\{c_i\}$ as hidden features

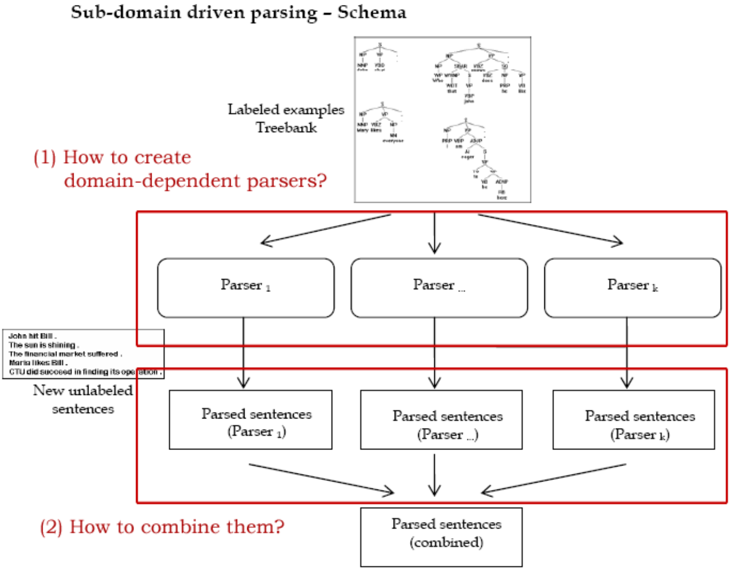
$$P(s,t) = \sum_i P(s,c_i)P(t|s,c_i) \tag{1}$$

This work: approximate it by creating an ensemble of parsers

Assumptions:

- We know a set of subdomains $\{c_i, \ldots, c_k\}$
- Approximate $\sum_i$ by combining predictions of subdomains parsers

Overview and Problem Statement

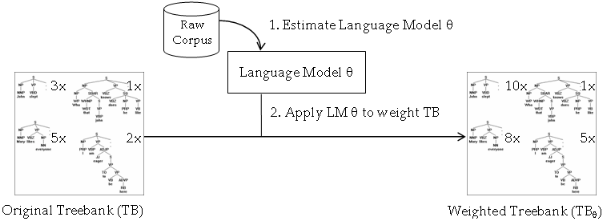


Creating subdomain-specific parsers

Weight the trees in treebank TB with subdomain statistics

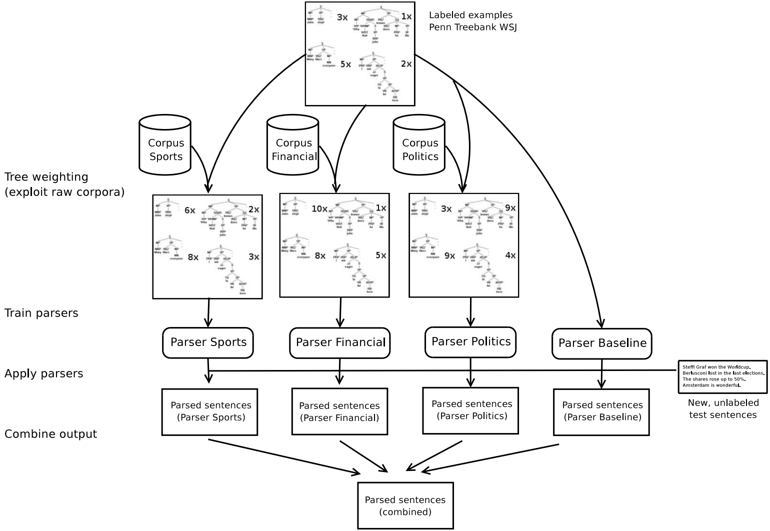
- Use domain-dependent raw corpus C (flat sentences)
- Induce statistical Language Model (LM) θ from C
- Assign a count f to every tree $\pi_i \in TB$ such that:

  f = average per-word "count" of yield $y_{[\pi_i]}$ under LM θ



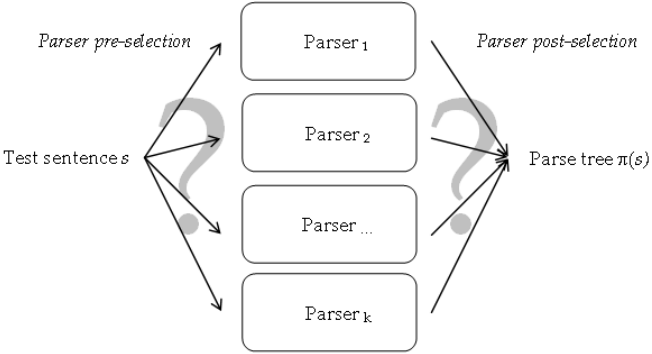
Retrain parser on subdomain-weighted TB_θ.

Overview of our approach - Details



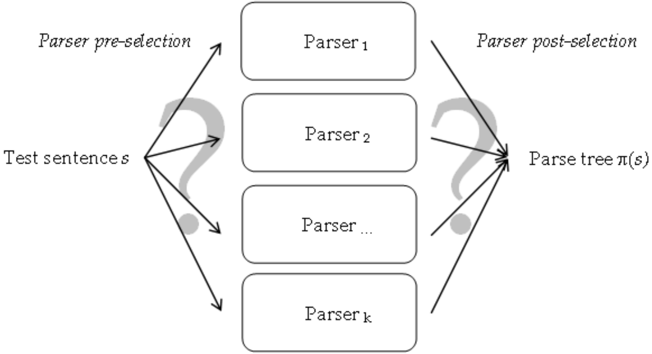
Parser Combination Techniques

How to combine them?



Parser Combination Techniques

How to combine them?



Parser Pre-selection: selecting a parser up-front (given: s)

Parser Post-selection: selecting a parser after parsing (given: s, t)

Pre-selection: Divergence Model (DVM)

We measure for every word how well it discriminates between the subdomains using the notion of divergence.

The *divergence* of a word w in a subdomain $i \in [1 \ldots k]$, from all other $(k-1)$ subdomains $(j \in [1 \ldots k], j \neq i)$:

$$divergence_i(w) = 1 + \frac{\sum_{j \neq i} |\log \frac{p_{\theta_i}(w)}{p_{\theta_j}(w)}|}{(k-1)} \tag{2}$$

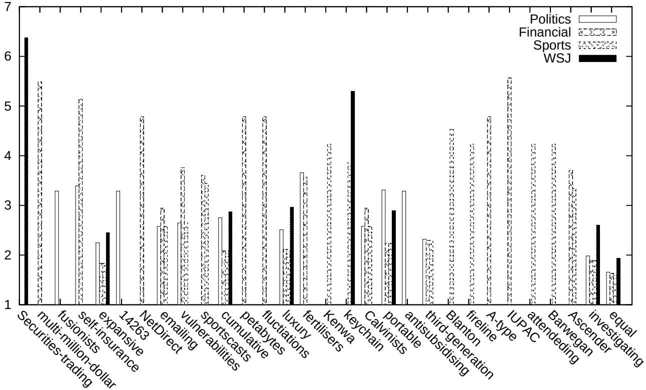
$$divergence\_sent_i(w_1^n) = \frac{\sum_{x=1}^{n} divergence_i(w_x)}{n} \tag{3}$$

Boundary issues:

- if $p_{\theta_i}(w) = 0$ then $divergence_i(w) = 1$, and
- if $p_{\theta_j}(w) = 0$, then $p_{\theta_j}(w) = 10^{-15}$ (constant).

Pre-selection: Divergence Model (DVM) - Example

For example, 'multi-million-dollar' (score FINANCIAL domain: 5.5), 'equal' (score all domains from 1.6 to 1.9)



Post-Selection: Node Weighting + DVM (NW-DVM)

For parse tree π_i with $1 \leq i \leq k$ and sentence w_1^n:

$$score(c) = \left[\frac{1}{k}\sum_{i=1}^{k}\delta[c, \pi_i]\right] \tag{4}$$

$$score(\pi_i) = (1-\lambda)\left[\frac{1}{|\pi_i|}\sum_{c\in\pi_i} score(c)\right] + \lambda * divergence\_sent_i(w_1^n) \tag{5}$$

where $|\pi_i|$ is the size of the constituent set, and $0 < \lambda < 1$ an interpolation factor.

- How well does the parse tree π_i fit the domain?
- How well does w_1^n fit the domain?

First Experiment: Variance among Parsers

- Are subdomain parsers complementary?
- Optimal decision procedure - an oracle:

$$\pi_{best\_oracle} = \mathrm{argmax}_i f_{\text{F-score}}(\pi_i) \tag{6}$$

First Experiment: Variance among Parsers

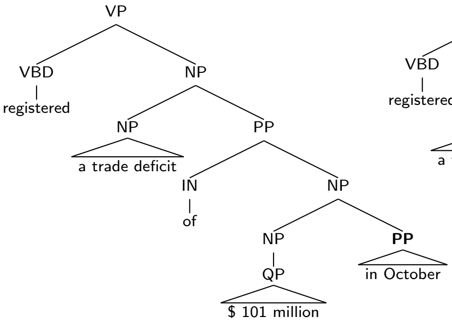
- Are subdomain parsers complementary?
- Optimal decision procedure - an oracle:

$$\pi_{best\_oracle} = \text{argmax}_i f_{\text{F-score}}(\pi_i) \qquad (6)$$

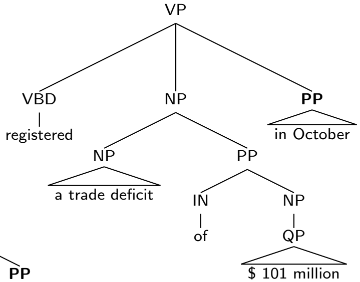
Parser	≤ 40 LR	LP	F-score
	Section 00 (development set)		
Baseline	89.44	89.63	89.53
Sports	88.95	88.83	88.89
Financial	89.01	88.84	88.92
Politics	88.86	88.70	88.78
Oracle combination	90.59	90.66	90.62
Improvement over baseline	+1.15	+1.03	**+1.09**
	Section 23 (test set)		
Baseline	88.77	88.87	88.82
Oracle combination	90.11	90.11	90.11
Improvement over baseline	+1.34	+1.24	**+1.29**

Effect Using Domain-awareness - Example

`Sent#90:` *South Korea registered a trade deficit of $ 101 million in October, reflecting the country's economic sluggishness, according to government figures released Wednesday.*



Parser$_{\mathrm{BASELINE}}$ F-score: 87.80%; incorrect PP-attachment



Oracle prediction F-score: 100% (*Parser*$_{\mathrm{FINANCIAL}}$ or *Parser*$_{\mathrm{POLITICS}}$)

Short Recap

- The example illustrates that a domain-specifically trained parser may find a correct or better result than the baseline parser.
- Our first experiment shows that our subdomain sensitive parsing instantiation in general has potential.
- We presented parser combination techniques that aim at achieving this potential.

Introduction and Motivation

Subdomain Sensitive Statistical Parsing

Subdomain Sensitive Parsers

Parser Combination Techniques

Experiments and Results

Conclusions and Future Work

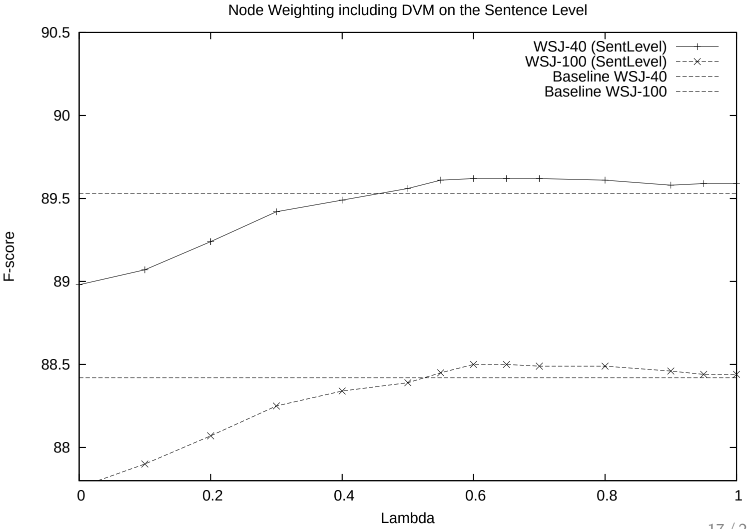
Results of Parser Combination Techniques

Parser	≤ 40 LR	≤ 40 LP	≤ 40 F-score
	Section 00 (development set)		
Baseline	89.44	89.63	89.53
Parser Pre-selection:			
Divergence Model (DVM)	89.50	89.68	**89.59**
Parser Post-selection:			
Node Weighting incl. DVM, $\lambda = 0.6$	89.53	89.71	**89.62**

Parser Post-selection NW-DVM highest F-score: 89.62%, i.e. $+0.09\%$ over baseline.

Results of Parser Combination Techniques

Result of Node Weighting incl. DVM (NW-DVM)



Results of Parser Combination Techniques

Summary

- Post-selection that considers both the parse tree and sentence performs best
- Nevertheless, it is closely followed by Parser Pre-selection based on the sentence only
- Results are confirmed on the test set (section 23):
  1. Node Weighting incl. DVM with $\lambda = 0.6$ (+0.08% F-score)
  2. Divergence Model (+0.03%)

Conclusions and Future Work

- Our first instantiation of subdomain sensitive parsing has indeed demonstrated to have potential
- However, combining the parsers to obtain a substantially better result is not an easy task
- Our approach leaves space open to extend, refine or improve various parts:
    - Other ways of instantiating domain-dependent parsers (e.g. self-training)
    - More sophisticated notion of domain
    - Further explore parser combination techniques
    - Explore to what extent *n*-best parsing might benefit from subdomain information

Introduction and Motivation

Subdomain Sensitive Statistical Parsing

Subdomain Sensitive Parsers

Parser Combination Techniques

Experiments and Results

Conclusions and Future Work

Thank you for your attention.

Treebank Weighting

Weight the trees in treebank TB with subdomain statistics and retrain parser.

- Use domain-dependent raw corpus C (flat sentences)
  - $C \in \{sports, financial, politics\}$
- Induce statistical Language Model (LM) θ from C
- Assign a count[a] f to every tree $\pi_i \in TB$:

$$f_\theta(\pi_i) = f_\theta(y_{[\pi_i]}) = -\log P_\theta(y_{[\pi_i]})/n \tag{7}$$

- Let f_θ^{max} be the maximum count of a tree in TB according to θ. The weight w_i assigned to π_i is defined as:

$$w_i = \text{round}\left\{\left(\frac{f_\theta^{max}}{f_\theta(\pi_i)}\right)^a\right\} \tag{8}$$

  where $a \geq 1$ is a scaling constant. In the default setting $a = 1$.

[a] f = average per-word "count" of the yield $y_{[\pi_i]}$ under LM θ