Incremental dependency-based parsing and bitext parsing

Matthias Buch-Kromann
and Martin Haulrich

TRE Workshop at Højstrupgård

27 January 2010

The problem: parsing and bitext parsing

Parsing

Compute the best syntax tree (parse) for a given sentence, ie, assign head word and dependency label to each word in the sentence.

Scoring model

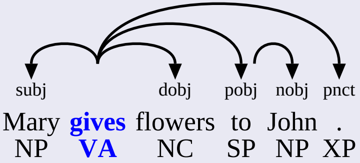
"Best" is defined in terms of some scoring model that can assign scores to parses and bitext parses, based on machine learning.

Bitext parsing

Compute the best bitext parse for a given bilingual sentence pair, ie, compute best combination of source tree, target tree, and alignment.

Dependency analyses

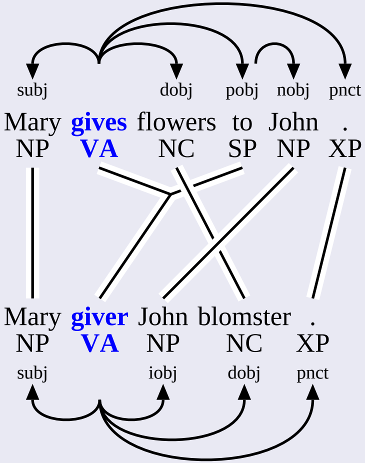
Dependency analysis: example



Bitext analysis

Source dependency analysis + target dependency analysis + word alignment.

Bitext analysis: example



Success criteria

High attachment score

Are the analyses produced by the model competitive with state-of-the-art in computational linguistics?

Computational efficiency

Is it possible to compute the analyses in a reasonable amount of time?

Linguistic expressiveness

Is the model rich enough from a linguistic perspective? Can it model the linguistic sophistication we can observe in humans?

Cognitive plausibility

Is it a plausible model of human language processing? In particular, is it incremental and non-monotonic?

How the problem relates to the four tracks

Translation systems (Track A)

Parsing and bitext parsing are key components in the dependency-based MT system we eventually want to build.

Translation products (Track B)

Parallel treebanks provide vital training data for our dependency parsers and bitext parsers.

Translation processes (Track C)

To model human reading processes and translation processes, we need cognitively plausible models of human parsing and bitext parsing. Also: incremental model could incorporate eye-movements.

Translation practices (Track D)

Might lead to better technology. Apart from that, no direct link to organizational and individual practices.

Incrementality

Incremental

Compute partial analyses as the input is received.

Humans.

Non-incremental

Do not compute anything before the entire input is received.

Most NLP systems. Eg, chart parsing, Minimum Spanning Tree parsing.

Monotonicity

Non-monotonic (repair-based)

Any aspect of the analysis can be revised at any point in the processing.

Humans.

Monotonic

Never change a decision once you have made it.

Most NLP systems.

Minimum-Spanning Tree parsing

MST parsing (McDonald et al)

Use minimum spanning-tree algorithm to compute non-projective parse. (Also projective variant with sibling information.)

Edge-factored scores

Scores are assigned to dependency edges without context. Eg: no complement frames and other edge-nonlocal information.

Properties

State-of-the-art performance. Non-incremental. Monotonic. Cannot be extended to non-edgefactored scores.

Can we keep the MST machine learner, but replace the MST parser with a better parser (incremental + non-monotonic + non-edgefactored)?

The incremental parser (unlabeled, edge-factored)

Incremental parsing (unlabeled)

N := words in sentence;
for $n = 1$ *to* N **do**
  AttachLeft(n);
  for $d = N - 1$ *to* 1 **do**
    if *n not dominated by d* **then**
      TryAttachRight(d,n)
    else
      TryAttachRightC(d,n)
return H

AttachLeft(n)

$H[n]$:= best-scoring left head

TryAttachRight(d, h)

if *h is a higher-scoring head for d* **then**
  H[d] := h

TryAttachRightCyclic(d,h)

foreach $p \in path(h,d) - \{d\}$ **do**
  foreach *node* h_p *not dominated by d* **do**
    if *higher score would result* **then**
      $H[d] := h$;
      $H[p] := h_p$

Results for unlabeled edge-factored parsing

Unlabeled attachment scores for CONLL-2006 data

	MST-trainer		IP-trainer			Search
	MST	IP	MST	IP	Loss	loss
danish	88.83	88.73	88.67	88.83	0.00	0.10
dutch	76.13	76.35	76.01	76.11	0.02	-0.22
german	85.80	85.40	85.32	85.44	0.36	0.40
portuguese	88.68	88.64	88.04	88.34	0.34	0.04
swedish	86.40	85.90	85.66	85.38	1.02	0.50

We are comparing with edge-factored MST parsing. 50k words for dutch, 100k words for other languages. (The reported loss may be smaller in reality because of choice of training iterations.)

Features beyond edge-factored models

Simple features

- Complement/adjunct counts
- Complement frame
- Left siblings and grandparents

Time-dependence

To guide the incremental parser, condition on timeline.1L

2L 4L ... EndL 1R 2R 4R ... EndR.

Sophisticated features

- Local word order (complement/adjunct siblings)
- Landing sites and temporary landing sites
- Worst-scoring edge on extraction path

Bitext parsing

Extending the algorithm to bitext parsing

Fundamentally same algorithm: read next word, try different repair operations until no improvements can be found, repeat.

Parsing operations

- monolingual operations
- alignment operations
- determining whether to read next word in source or target text

Scoring features

- monolingual features
- word alignment features
- features relating alignments to dependency structure

Incremental parsing: advantages and disadvantages

Incremental parsing: advantages

- incremental, non-monotonic
- psycholinguistic plausibility
- non-edgefactored features (may outweigh loss from search error)
- easy to extend to bitext parsing in principle

Incremental parsing: the price

- slightly smaller attachment score because of search error
- slightly slower than MST (perhaps much slower in full parser)

Conclusion

Conclusion

Incremental non-monotonic parsing seems feasible.

The approach needs to be tried.

Interesting both in terms of NLP and psycholinguistics.