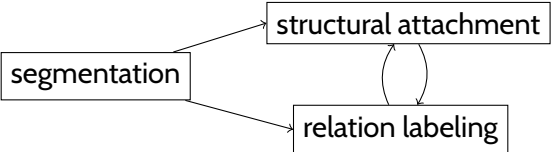


Greedy bottom up RST parsing

Antoine Venant

November 6, 2017

Last sessions recap

- ▶ Generic process: 

```
graph LR; A[segmentation] --> B[structural attachment]; A --> C[relation labeling]; B <--> C
```
- ▶ Different theories, different structures and interpretations.
- ▶ Different corresponding datasets.
- ▶ Difficult problem: mix of lexical, syntactic, semantic knowledge and contextual reasoning.

Today

- ▶ Framework of choice: RST.
- ▶ Data: (binarized) RST Treebank with coarse-grained relations.
- ▶ Given what we now know of RS Trees: How would you design a parser?

Hilda

First look at a *complete* text-level parser:

[Hernault & all, 2010]

- ▶ Hernault, H., Prendinger, H. and Ishizuka, M., 2010. HILDA: A discourse parser using support vector machine classification. *Dialogue & Discourse*, 1(3).
- ▶ Experiments with a refined set of features: Feng, V.W. and Hirst, G., 2012, July. Text-level discourse parsing with rich linguistic features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1* (pp. 60-68). Association for Computational Linguistics.
- ▶ Not the first discourse parser (e.g. mostly rule-based Soricut & Marcu 2003).
- ▶ But one of the first statistical (supervised learned) parser to build complete structure accross sentences.

Segmentation

- ▶ Segmentation as a separate, preliminary task.
- ▶ Idea: classify each word either as *boundary* 1 or *intermediate* 0.
- ▶ Turn sequence of words into sequence of k -dim feature vectors:
 $\mathcal{W}^n \mapsto (\mathbb{R}^k)^n$.
- ▶ Each words of the training set becomes a training instance.
- ▶ Learn $\mathcal{S} : \mathbb{R}^k \mapsto \{0, 1\}$.
- ▶ Hilda models this step with max a margin model (SVM). We won't detail training algorithm.
- ▶ Huge majority of 0's instances in the training data.

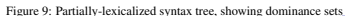
Example from Hernault & all

[Farm₀ lending₀ was₀ enacted₁] [to₀ correct₀ this₀ problem₁] [by₀
providing₀ a₀ reliable₀ flow₀ of₀ lendable₀ funds₁]

Features for segmentation

- ▶ Lexico-Syntactic features from [Soricut & Marcu, 2003].
- ▶ Extracted from *lexicalized* parse tree.
- ▶ Project lexical *heads* from every node of parse tree using predefined rule (e.g., [Magermann, 1995])
- ▶ For word w_i , find highest ancestor N_i with w_i as lexical head.
- ▶ Find P_i and R_i resp. N_i 's parent and right sibling.
- ▶ Contextual feature at index i : P_i, N_i, R_i + resp. POS tags and lexical heads for those.
- ▶ Features for w_i : concatenation of contextual features at indices $i - 2, i - 1, i$.

Picture from [Hernault & all 2010]



Segmenter performance

- ▶ 95% F-Score using penn treebank gold parse trees.
- ▶ 94% F-Score using external parser.
- ▶ Human annotators' agreement at 98%

Parsing as search

After segmentation: new input is a sequence e_1, \dots, e_n of EDUs, with $e_i = w_0^i \dots w_{|e_i|}^i$

- ▶ Think of parsing as a search (or optimisation) problem.
- ▶ Among a set of possible discourse structure find the one with best *score*.
- ▶ Far too many possible structure for brute force.
- ▶ Also, how do we assign score to a given structure?

Parsing as search contd

- ▶ One possibility: use a scoring function which ‘decomposes’ locally over the input.
- ▶ Learn classifier assigning scores to *local* structural choices.
- ▶ *Decode* from local scores assignments into global structure.

However

- ▶ Some local structural choices might prevent others.
- ▶ e.g., in RST, $R(e_1, e_3)$ prevents $R'(e_2, e_4)$.
- ▶ Still a search problem.

Exploitable properties of RS Trees

Projetivity: Relations hold only between adjacent spans. → **bottom-up construction**

Nuclearity Principle: Relations holding between complex span must hold *a minima* between their most salient part. → we'll come back to this.

Greedy bottom up parsing

- ▶ Relax search for optimal structure.
- ▶ Make locally optimal choices in sequence.
- ▶ Make use of RST projectivity:
 - ▶ At elementary level connect only adjacent *EDUs*.
 - ▶ Replace connected edus with total covered *span* in input.
 - ▶ At every level, connect only adjacent spans.

The algorithm – classifiers

- ▶ Set of relation labels $R_L = \{\text{attribution, elaboration, } \dots\}$.
- ▶ Set of labeling decision: $J : R_L \times \{(S, N), (N, S), (N, N)\}$ (label + nuclearity).
- ▶ Use two classifiers: \mathcal{A} and \mathcal{L} .
- ▶ At every step input is a sequence of subtrees t_1, \dots, t_n
- ▶ and the l -dimensional vector encoding of contiguous spans t_i, t_{i+1} :
 $v_1, \dots, v_{n-1} \in (\mathbb{R}^l)^{n-1}$.
- ▶ $\mathcal{A} : \mathbb{R}^l \mapsto [0, 1]$
- ▶ multi-class classifier $\mathcal{L} : \mathbb{R}^l \mapsto J$
- ▶ Each span in training set gives rise to a training instance (corresponding to encompassed relation between direct subspans).

The algorithm – parsing

Input: a sequence t_1, \dots, t_n of RS Trees.

1. If $n = 1$ returns t_1
2. Else compute v_1, \dots, v_{n-1} the encoding sequence of l -dimensional feature vectors for each input pair (t_i, t_{i+1}) .
3. For every i in $[1, n - 1]$, compute $s_i = \mathcal{A}(v_i)$
4. Find $i^* = \operatorname{argmax}_i(s_i)$
5. Compute $(r, (X, Y)) = \mathcal{L}(v_{i^*}, v_{i^*+1})$ (recall: label + nuclearity).
6. Let $t'_1, \dots, t'_{n-2} = t_1, \dots, t_{i^*-1}, r(t_{i^*} _ X, t_{i^*+1} _ Y), t_{i^*+2} \dots t_n$.
7. Return the result of recursive application to the t' sequence.

Remark:

- ▶ No guarantee to find optimal structure w.r.t. sum of local scores.
- ▶ Runtime $O(n)!$

Features related to text organisation

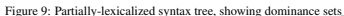
- ▶ For some relations (e.g., contrast), satellite are generally shorter than nuclei.

Belong to same sentence	Common
Belong to same paragraph	Common
Number of paragraph boundaries	Each
Number of sentence boundaries	Each
Length in tokens	Each
Length in EDUs	Each
Distance to beginning of sentence in tokens	Each
Size of span over sentence in EDUs	Each
Size of span over sentence in tokens	Each
Size of both spans over sentence in tokens	Common
Distance to beginning of sentence in EDUs	Each
Distance to beginning of text in tokens	Each
Distance to end of sentence in tokens	Each

Discourse cues

- ▶ Could use dictionary approach with connectives.
- ▶ But doesn't capture non-lexicalized cues.
- ▶ instead, use N-Grams (trigrams).
- ▶ Idea: cues toward the boundaries.
- ▶ Use trigram at beginning and end of each edus ($2*2=4$ trigram per instance). + POS tags for span's prefix and suffix.
- ▶ Tested against dictionary (see paper).

Picture from [Hernault & all 2010]



Nuclearity principle

- ▶ Project spans to their salient part.
- ▶ Salient part of an EDU e is itself.
- ▶ Salient part of an NS span is its nucleus.
- ▶ Salient part of an NN (left resp. right) span is its (right resp. left) nucleus.
- ▶ Add to feature vector for (t_1, t_2) the feature vector for the salient parts of t_1 (left) and t_2 respectively.

Encoding recursive structure

- ▶ Writes down breadth-first traversal string representation of binary tree (up to depth 3—empirical finding).

Evaluation and discussion

- ▶ Project paper's section.