Ten (Or More) Minutes on Unsupervised Parsing

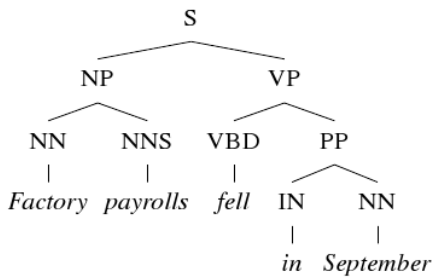
Franco M. Luque

Outline

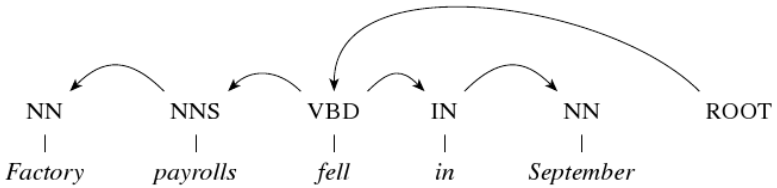
The Problem of Parsing

Obtain the syntactic structure of a sentence. The two most common linguistic structures are:

- Constituent trees:



- Dependency structures:



- These usually include word categories, called POS tags.

The Problem of Supervised Parsing

Supervised Parsing is the problem of building a parser using a treebank.

- Treebanks are corpuses of parsed sentences.
- A part of the treebank is used to train the parser.
- Another part is used to evaluate the parser.

It can be seen as learning a function $f : X \rightarrow Y$ (the parser) by knowing some points $(x_1, f(x_1)), \ldots, (x_n, f(x_n))$ (the treebank).

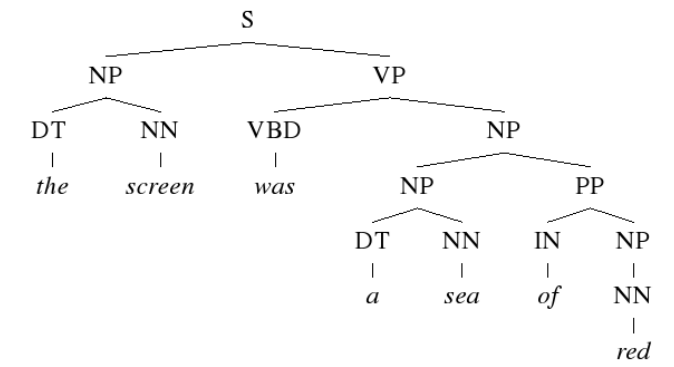
The Problem of Unsupervised Parsing

What can we do by only knowing a set of points $\{x_1, \ldots, x_n\}$ of the domain of f?

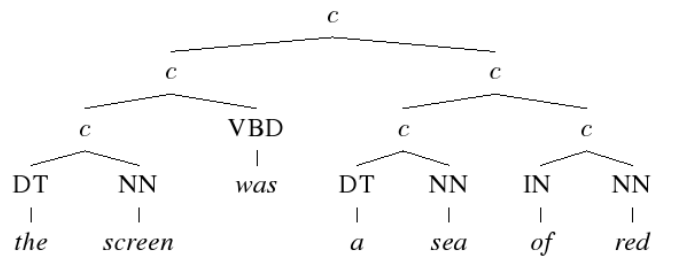
- The parser is trained with sentences.
- The evaluation is still done against a treebank.
- The set of syntactic categories is unknown.

Parser Evaluation

Gold Tree:



Proposed Tree:



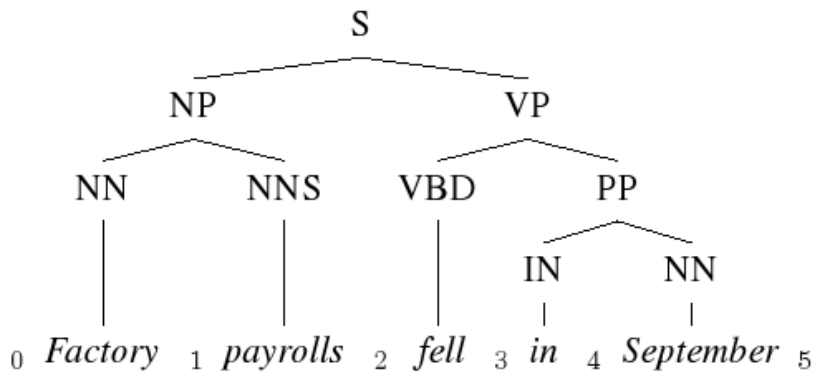
- Precision: proportion of constituents that are right (4/5).
- Recall: proportion of right constituents that are found (4/5).
- F1: Harmonic mean between Precision and Recall.

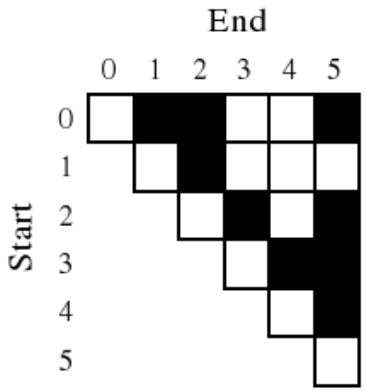
There are several variants of this measures.

The DMV+CCM Model

- Developed by Dan Klein and Chris Manning in 2004.
- Parses dependency trees that can be converted to binary bracketings.
- Learn and parse from POS tags instead of words. Must be combined with a POS tagger to obtain a real parser.
- Evaluated on English, German and Chinese treebanks, only with sentences of length ≤ 10.
- Punctuation is not considered in order to emulate spoken language.

Constituents and Contexts





Span	Label	Constituent	Context
⟨0,5⟩	S	NN NNS VBD IN NN	◇ – ◇
⟨0,2⟩	NP	NN NNS	◇ – VBD
⟨2,5⟩	VP	VBD IN NN	NNS – ◇
⟨3,5⟩	PP	IN NN	VBD – ◇
⟨0,1⟩	NN	NN	◇ – NNS
⟨1,2⟩	NNS	NNS	NN – VBD
⟨2,3⟩	VBD	VBD	NNS – IN
⟨3,4⟩	IN	IN	VBD – NN
⟨4,5⟩	NN	NNS	IN – ◇

The Constituent Context Model

- Parses binary bracketings (unlabeled binary trees).
- It is a generative model (defines $P(s, B)$ in terms of $P(s|B)$):

$$P(s, B) = P_{bin}(B)P(s|B)$$

- Each span ${}_is_j$ and context s_{i-1}, s_j is generated with probability conditioned on the boolean B_{ij}:

$$P(s|B) = \prod_{i,j:i\leq j} P_{span}({}_is_j|B_{ij})P_{ctx}(s_{i-1}, s_j|B_{ij})$$

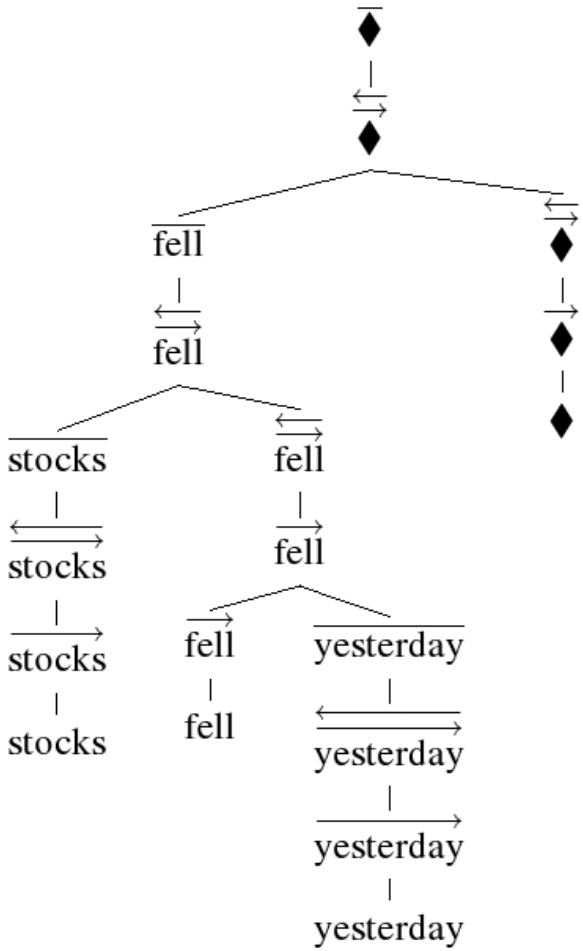
- Obviously, this is mass deficient.

CCM Training

- The parameters of the model are the values of P_{span} and P_{ctx}.
- The model is trained with a corpus S of sentences (of POS tags).
- Iterative Expectation Maximization is used to maximize the likelihood $\prod_{s \in S} P(s)$.

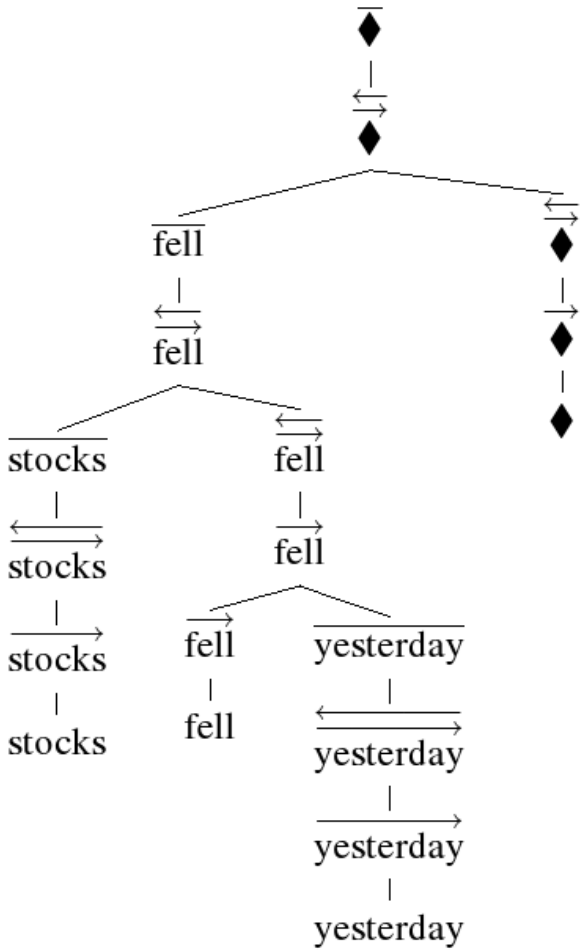
The Dependency Model with Valence

- Parses dependency trees that can be converted to binary bracketings.
- The dependency structure is generated from the sentence in a head-outwards procedure.
- A word takes dependents until it decides to stop, doing this to each side independently.



DMV Training

- Each decision in the procedure has a probability. These are the parameters of the model.
- The probability $P(s, D)$ is the product of the probabilities of the decisions taken to build D.
- Iterative Expectation Maximization is used to maximize the likelihood $\prod_{s \in S} P(s)$.



The CCM and DMV Models Combined

- Parses the same structures as DMV.
- The generation procedure includes the CCM factor.

Results:

- The strengths of both models are complementary.
- Combined with a tagger, performance drops, but not below CCM's.

	Constituency			Dependency	
Model	UP	UR	UF_1	Dir	Undir
English (WSJ10 – 7422 Sentences)					
LBRANCH/RHEAD	25.6	32.6	28.7	33.6	56.7
RANDOM	31.0	39.4	34.7	30.1	45.6
RBRANCH/LHEAD	55.1	70.0	61.7	24.0	55.9
DMV	46.6	59.2	52.1	43.2	62.7
CCM	64.2	81.6	71.9	23.8	43.3
DMV+CCM (POS)	**69.3**	**88.0**	**77.6**	**47.5**	**64.5**
DMV+CCM (DISTR.)	65.2	82.8	72.9	42.3	60.4
UBOUND	78.8	100.0	88.1	100.0	100.0
German (NEGRA10 – 2175 Sentences)					
LBRANCH/RHEAD	27.4	48.8	35.1	32.6	51.2
RANDOM	27.9	49.6	35.7	21.8	41.5
RBRANCH/LHEAD	33.8	60.1	43.3	21.0	49.9
DMV	38.4	69.5	49.5	40.0	57.8
CCM	48.1	85.5	61.6	25.5	44.9
DMV+CCM	**49.6**	**89.7**	**63.9**	**50.6**	**64.7**
UBOUND	56.3	100.0	72.1	100.0	100.0
Chinese (CTB10 – 2437 Sentences)					
LBRANCH/RHEAD	26.3	48.8	34.2	30.2	43.9
RANDOM	27.3	50.7	35.5	35.9	47.3
RBRANCH/LHEAD	29.0	53.9	37.8	14.2	41.5
DMV	**35.9**	**66.7**	**46.7**	42.5	54.2
CCM	34.6	64.3	45.0	23.8	40.5
DMV+CCM	33.3	62.0	43.3	**55.2**	**60.3**
UBOUND	53.9	100.0	70.1	100.0	100.0

Why Use Punctuation?

- Punctuation gives syntactic information that must be used.
- This information is more important as sentences are longer.
- These claims have been verified in other models for Unsupervised Parsing (Seginer 2007).

How Can We Use Punctuation?

- Opening/closing punctuation determines brackets with prob. 1 (e.g. parenthesis and quotes).
- From individual punctuation there are no language independent rules. It is modeled into DMV parameters and it's behaviour is learnt.
- This way we could improve DMV+CCM performance from 77.6 to 78.4.

Possible Future Work

- Combine unsupervised POS tagging and unsupervised parsing.
- Develop a dependency parser with a faster training algorithm.
- Parse longer sentences.
- Study syntactic category induction.