Statistical Parsing

Statistical context-free parsing

Çağrı Çöltekin

University of Tübingen
Seminar für Sprachwissenschaft

November 15, 2016

Ingredients of a (natural language) parser

- A grammar
- An algorithm for parsing
- A method for ambiguity resolution

Context free grammars

- Context free grammars are adequate for expressing most phenomena in natural language syntax
- Most of the parsing theory (and practice) is build on parsing CF languages
- The context-free rules have the form

$$A \to \alpha$$

  where A is a single non-terminal symbol and α is a (possibly empty) sequence of terminal or non-terminal symbols
- We will mainly focus with parsing with context-free grammars for the rest of this lecture

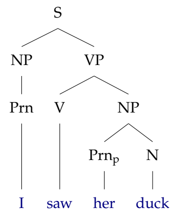
Parsing with context-free grammars

- Parsing can be
  - top down: start from S, search for derivation that leads to the input
  - bottom up: start from input, try to *reduce* it to S
- Naive search for both recognition/parse is intractable
- Dynamic programming methods allow polynomial time *recognition*

CKY bottom-up, requires Chomsky normal form
Earely top-down (with bottom-up filtering), works with unrestricted grammars
  - $O(n^3)$ time complexity (for recognition)

Representations for a parse

A parse tree:



A history of derivations:

- $S \Rightarrow NP\ VP$
- $NP \Rightarrow Prn$
- $Prn \Rightarrow I$
- $VP \Rightarrow V\ NP$
- $V \Rightarrow saw$
- $NP \Rightarrow Prn_p\ N$
- $Prn_p \Rightarrow her$
- $N \Rightarrow duck$

A sequence with (labeled) brackets

$$\Big[_S \big[_{NP} [_{Prn}\ I] \big] \big[_{VP} [_V\ saw] \, [_{NP} [_{Prn_p}\ her] \, [_N\ duck]] \big] \Big]$$

Chart parsing example (CKY recognition)

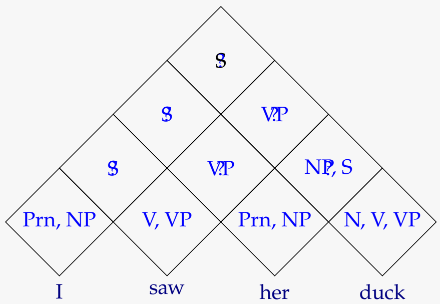
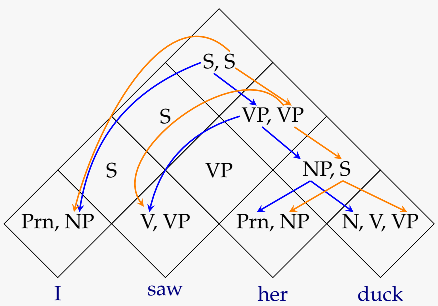


Chart parsing example (CKY parsing)



CF chart parsing

- With chart parsing, we can get polynomial recognition complexity (recovering all parses from the chart may still require exponential time)
- The chart parser also store multiple parses (the resulting *parse forest*) in an efficient way
- But the methods that we discussed so far cannot help us resolve ambiguity

Pretty little girl's school (again)

Cartoon Theories of Linguistics, SpecGram Vol CLIII, No 4, 2008. `http://specgram.com/CLIII.4/school.gif`

Some more examples

- Lexical ambiguity
  - She is looking for a match
  - We saw her duck
- Attachment ambiguity
  - I saw the man with a telescope
  - Panda eats bamboo shoots and leaves
- Local ambiguity (garden path sentences)
  - The horse raced past the barn fell
  - The old man the boats
  - Fat people eat accumulates
- Anaphora resolution
  - Every farmer who owns a donkey beats it.

Even more examples

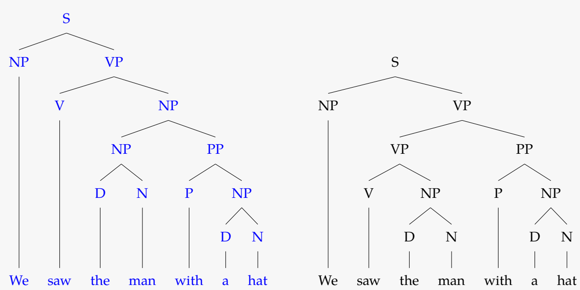
(newspaper headlines)

- FARMER BILL DIES IN HOUSE
- TEACHER STRIKES IDLE KIDS
- SQUAD HELPS DOG BITE VICTIM
- BAN ON NUDE DANCING ON GOVERNOR'S DESK
- PROSTITUTES APPEAL TO POPE
- KIDS MAKE NUTRITIOUS SNACKS
- DRUNK GETS NINE MONTHS IN VIOLIN CASE
- MINERS REFUSE TO WORK AFTER DEATH

But humans do not recognize many ambiguities

- Time flies like an arrow; fruit flies like a banana
- Outside of a dog, a book is a man's best friend; inside it's too hard to read
- One morning I shot an elephant in my pajamas. How he got in my pajamas, I don't know.
- Don't eat the pizza with a knife and fork

The task: choosing the most plausible parse



Statistical parsing

- Find the most plausible parse of an input string given all possible parses
- We need a scoring function, for each parse, given the input
- We typically use probabilities for scoring, task becomes finding the parse (or tree), t, given the input string x

$$t_{best} = \arg\max_t P(t|x)$$

- Note that some ambiguities need a larger context than the sentence to be resolved correctly

Probability refresher (1)

- Probability is a measure of (un)certainty of an event
- We quantify the probability of an event with a number between 0 and 1
  - 0 the event is impossible
  - 0.5 the event is as likely to happen (or happened) as it is not
  - 1 the event is certain
- All possible outcomes of a trial (experiment or observation) is called the *sample space* (Ω)

Axioms of probability states that

1. $P(E) \in \mathbb{R}$, $P(E) \geq 0$
2. $P(\Omega) = 1$
3. For *disjoint* events E_1 and E_2, $P(E_1 \cup E_2) = P(E_1) + P(E_2)$

Probability refresher (2)

Joint and conditional probabilities, chain rule

- Joint probability of two events is noted as $P(x, y)$
- The conditional probability is defined as

$$P(x|y) = \frac{P(x,y)}{P(y)} \text{ or } P(x, y) = P(x|y)P(y)$$

- If the events x and x are independent,

$$P(x|y) = P(x), P(y|x) = p(y), P(x, y) = P(x)P(y)$$

- For more than two variables (chain rule):

$$P(x, y, z) = P(z|x, y)P(y|x)P(x) = P(x|y, z)P(y|z)P(z) = \ldots$$

- If all are independent

$$P(x, y, z) = P(x)P(y)P(z)$$

Probabilistic context free grammars (PCFG)

A probabilistic context free grammar is specified by,

- Σ is a set of terminal symbols
- N is a set of non-terminal symbols
- $S \in N$ is a distinguished *start* symbol
- R is a set of rules of the form

$$A \rightarrow \alpha \quad [p]$$

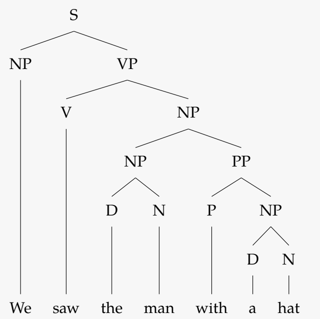
where A is a non-terminal, α is string of terminals and non-terminals, and p *is the probability associated with the rule*

- The grammar accepts a sentence if it can be derived from S with rules $R_1 \ldots R_k$
- *The probability of a parse* t *of input string* x, $P(t|x)$, *corresponding to the derivation* $R_1 \ldots R_k$ *is*

$$P(t|x) = \prod_1^k p_i$$

where p_i *is the probability of the rule* R_i

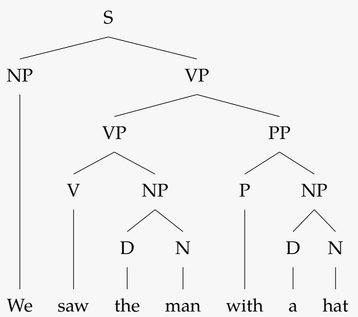
PCFG example (1)



S	$\rightarrow$ NP VP	1.0
NP	$\rightarrow$ D N	0.7
NP	$\rightarrow$ NP PP	0.2
NP	$\rightarrow$ We	0.1
VP	$\rightarrow$ V NP	0.9
VP	$\rightarrow$ VP PP	0.1
PP	$\rightarrow$ P NP	1.0
N	$\rightarrow$ hat	0.2
N	$\rightarrow$ man	0.8
V	$\rightarrow$ saw	1.0
P	$\rightarrow$ with	1.0
D	$\rightarrow$ a	0.6
D	$\rightarrow$ the	0.4

$$P(t) = 1.0 \times 0.1 \times 0.9 \times 1.0 \times 0.2 \times 0.7 \times 0.4 \times 0.8 \times 1.0 \times 1.0 \times 0.7 \times 0.6 \times 0.2$$
$$= 0.000263424$$

PCFG example (2)



S	$\rightarrow$ NP VP	1.0
NP	$\rightarrow$ D N	0.7
NP	$\rightarrow$ NP PP	0.2
NP	$\rightarrow$ We	0.1
VP	$\rightarrow$ V NP	0.9
VP	$\rightarrow$ VP PP	0.1
PP	$\rightarrow$ P NP	1.0
N	$\rightarrow$ hat	0.2
N	$\rightarrow$ man	0.8
V	$\rightarrow$ saw	1.0
P	$\rightarrow$ with	1.0
D	$\rightarrow$ a	0.6
D	$\rightarrow$ the	0.4

$$P(t) = 1.0 \times 0.1 \times 0.3 \times 0.7 \times 1.0 \times 0.1 \times 0.8 \times 0.4 \times 0.8 \times 1.0 \times 1.0 \times 0.7 \times 0.6 \times 0.2$$
$$= 0.0001317120$$

Where does the rule probabilities come from?

- Supervised: estimate from a treebank, e.g., using maximum likelihood estimation
- Unsupervised: expectation-maximization (EM)

PCFGs - an interim summary

- PCFGs assign probabilities to parses based on CFG rules used during the parse
- PCFGs assume that the rules are independent
- PCFGs are generative models, they assign probabilities to $P(t, x)$, we can calcuate the probability of a sentence by

$$P(x) = \sum_t P(t, x) = \sum_t P(t)$$

What makes the difference in PCFG probabilities?

S	$\Rightarrow$ NP VP	1.0	S	$\Rightarrow$ NP VP	1.0
NP	$\Rightarrow$ We	0.1	NP	$\Rightarrow$ We	0.1
VP	$\Rightarrow$ VP PP	0.1	VP	$\Rightarrow$ V NP	0.7
VP	$\Rightarrow$ V NP	0.8	V	$\Rightarrow$ saw	1.0
V	$\Rightarrow$ saw	1.0	NP	$\Rightarrow$ NP PP	0.2
NP	$\Rightarrow$ D N	0.7	NP	$\Rightarrow$ D N	0.7
D	$\Rightarrow$ the	0.4	D	$\Rightarrow$ the	0.4
N	$\Rightarrow$ man	0.8	N	$\Rightarrow$ man	0.8
PP	$\Rightarrow$ P NP	1.0	PP	$\Rightarrow$ P NP	1.0
P	$\Rightarrow$ with	1.0	P	$\Rightarrow$ with	1.0
NP	$\Rightarrow$ D N	0.7	NP	$\Rightarrow$ D N	0.7
D	$\Rightarrow$ a	0.6	D	$\Rightarrow$ a	0.6
N	$\Rightarrow$ hat	0.2	N	$\Rightarrow$ hat	0.2

The parser's choice would not be affected by lexical items!

What is wrong with PCFGs?

- In general: the assumption of independence
- The parents affect the correct choice for children, for example, in English NP $\rightarrow$ Prn is more likely in the subject position
- The lexical units affect the correct choice decision, for example:
  - We eat the pizza with hands
  - We eat the pizza with mushrooms
- Additionally: PCFGs use local context, difficult to incorporate arbitrary/global features for disambiguation

Solutions to PCFG problems

- Independence assumptions can be relaxed by either
  - Parent annotation
  - Lexicalization - Collins (1999)
- To condition on arbitrary/global information: disciriminative models - Charniak and Johnson (2005)

Evaluating the parser output

- A parser can be evaluated
  - extrinsically: based on it's effect on a task (e.g., machine translation) where it is used
  - intrinsically: based on the match with ideal parsing
- The typically evaluation (intrinsic) based on a *gold standard* (GS)
- Exact match is often
  - very difficult to achieve (think about a 50-word newspaper sentence)
  - not strictly necessary (recovering parts of the parse can be useful for many purposes)

Parser evaluation metrics

- Common evaluation metrics are (PARSEVAL):
  - precision: the ratio of correctly predicted nodes
  - recall: the nodes (in GS) that are predicted correctly
  - f-measure: harmonic mean of precision and recall $\left(\frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}\right)$
- The measures can be
  - unlabled: the spans of the nodes are expected to match
  - recall: the node label should also match
- Crossing brackets (or average non-crossing brackets)
  (We (saw (them (with binoculars))))
  (We ((saw them) (with binoculars)))
- Measures can be averaged per constituent (micro average), or over sentences (macro average)

Training, test, development sets

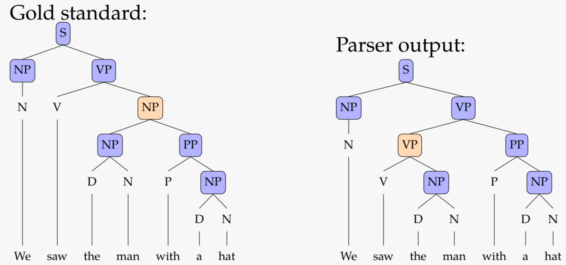
You already know it, but to be sure …

- Testing a statistical (machine learning) model on the training set is cheating (or fooling yourself)
- The systems has to be tested on a separate *test set*
- We often need to fine-tune the model, adjust parameters based on its performance on a *development set*
- Actual training is carried over on a *training set*
- One should also follow the same ideas when using cross-validation

PARSEVAL example

Gold standard:

Parser output:



$$\text{precision} = \frac{6}{7} \quad \text{recall} = \frac{6}{7} \quad \text{f-measure} = \frac{6}{7}$$

Problems with PARSEVAL metrics

- PARSEVAL metrics favor certain type of structures
  - You can surprisingly do well for flat tree structures (e.g., Penn treebank)
  - Results of some mistakes are catastrophic (e.g., low attachment)
- Not all mistakes are equally important for semantic distinctions
- Some alternatives:
  - Extrinsic evaluation
  - Evaluation based on extracted dependencies

Summary

- PCFGs are a good first start for statistical parsing
- But they are limited (mainly due to independence assumption)

Next week: (statistical) dependency parsing
Please read: Joakim Nivre (n.d.). *Dependency grammar and dependency parsing*. Unpublished notes. URL: http://stp.lingfil.uu.se/~nivre/docs/05133.pdf

Bibliography

Charniak, Eugene and Mark Johnson (2005). "Coarse-to-fine N-best Parsing and MaxEnt Discriminative Reranking". In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. ACL '05. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 173–180. DOI: 10.3115/1219840.1219862. URL: http://dx.doi.org/10.3115/1219840.1219862.

Collins, Michael (1999). "Head-Driven Statistical Models for Natural Language Parsing". PhD thesis. University of Pennsylvania.

Nivre, Joakim (n.d.). *Dependency grammar and dependency parsing*. Unpublished notes. URL: http://stp.lingfil.uu.se/~nivre/docs/05133.pdf.