Probabilistic Parsing

M. Rajman & J.-C. Chappelier

Laboratoire d'Intelligence Artificielle
Faculté I&C

Objectives and contents of this lecture

Objectives:

➥ Present Stochastic Context-Free Grammars (SCFGs),
a probabilistic extension of CFGs to make choices among parse trees

Contents:

① Introduction: probabilities
- Why?
- How?
- What?

② *n*-grams

③ SCFGs
- Introduction / Notations
- Definition
- Learning

Parsing: probabilistic approach: why

WHY probabilities? (at the syntactic level)

Linguistic resources needed for semantic/pragmatic models,
even for more sophisticated syntactic models,
are hard to obtain/create

☞ **Extension** of (simple) standard syntactic models

☞ to be able to **make choices** among sentences/structures (in case of ambiguity)

☞ Automatic **Learning** of models from corpora

M. Rajman & J.-C. Chappelier
EPFL

Parsing: probabilistic approach: how

What does it mean to "probabilize"?

☞ Implicitly represent the linguistic constraints that we do not want to or do not know how to integrate into the models:

Set of linguistic phenomena that **cannot** or are **hard to express** in operational terms but that still are **possible to evaluate** (on corpora)

The probability is then a measure of the quality of the adequation between the sentence/structure and the underlying model

M. Rajman & J.-C. Chappelier

Parsing: probabilistic approach: what (1/3)

WHAT is "probabilized"?

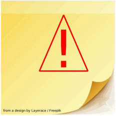
☞ The point of view is different depending on whether the syntactic model is used as a **recognizer** or as an **analyzer**

Reminder:

- A *recognizer* in only able to tell whether the input sentence is correct or not.
- An *analyzer* is more complex and produces additional information for the correct sentences: a structure representing the syntactic organization of the words.

M. Rajman & J.-C. Chappelier
EPFL

Parsing: probabilistic approach: what (2/3)



	recognizer	analyzer
what is probabilized?	sentences	parse trees associated to a given sentence
meaning of the probabilities	adequation of a sentence to the model $P(w_1^n)$	adequation of a structure (tree) to the model $P(T \mid w_1^n)$
example	N-grams	SCFG

Notice: Although in principle probabilities have no reason to depend on the formal description of the language they are associated with, their operational definition in practice can hardly be built independently of the generative model defining the language (i.e. the grammar)

Parsing: probabilistic approach: what (3/3)

General scheme of realization of probabilistic model:

- Identify the probability to estimate: $P(W_1 ... W_n)$ or $P(T | W_1 ... W_n)$
- On the basis of linguistic hypotheses, express this probability by a restricted number of parameters: $P = f(p_1 ... p_k)$
- On the basis of a well defined corpora, estimate the parameters in order to be able to compute probabilities

M. Rajman & J.-C. Chappelier

N-grams (reminder)

One possible probabilization of a language: probabilities of *fixed-size sequences* of words (N-grams of words) and then approximate the probabilities of a longer sequence on the basis of these parameters:

$$P(w_1,\ldots,w_n) = P(w_1,\ldots,w_N) \cdot \prod_{i=N+1}^{n} P(w_i|w_{i-N+1},\ldots,w_{i-1})$$

Examples ($N = 2$):

the cat ate a mouse	ate mouse a cat the
(the cat) (cat ate) (ate a) (a mouse)	(ate mouse) (mouse a) (a cat) (cat the)

❢ For an accurate estimation, **huge** amounts of data are required (+ smoothing)

M. Rajman & J.-C. Chappelier
EPFL

SCFG definition

a Stochastic Context-Free Grammar (SCFG) is

- a CFG for which
- each rule R is associated with a stochastic coefficient $p(R)$ such that
  - $0 \leq p(R) \leq 1$
  - $\displaystyle\sum_{R':\text{left}(R')=\text{left}(R)} p(R') = 1$
- $P(T = R_1 \circ \ldots \circ R_n) = \displaystyle\prod_{i=1}^{n} p(R_i)$

Maximization or consistent grammars

A simplified example of a SCFG (1/2)

From the last lesson:

syntactic rules:

R_1:	S	$\rightarrow$ NP VP		(p_1)
R_2:	VP	$\rightarrow$ V		(p_2)
R_3:	VP	$\rightarrow$ V NP		(p_3)
R_4:	NP	$\rightarrow$ Det N		(p_4)

lexical rules:

L_1:	N	$\rightarrow$ cat	(q_1)
L_2:	Det	$\rightarrow$ the	(q_2)
L_3:	Det	$\rightarrow$ a	(q_3)
L_4:	N	$\rightarrow$ mouse	(q_4)
L_5:	V	$\rightarrow$ ate	(q_5)

with:

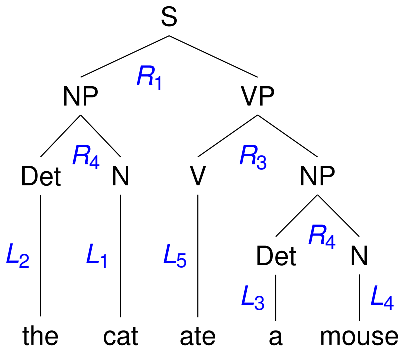
$$p_1 = 1$$
$$p_2 + p_3 = 1$$
$$p_4 = 1$$

$$q_1 + q_4 = 1$$
$$q_2 + q_3 = 1$$
$$q_5 = 1$$

Notice how lexical rules probabilities relates to emission probabilities of HMMs for PoS tagging.

A simplified example of a SCFG (2/2)

$P($



$) = p_1 \cdot p_4 \cdot q_2 \cdot q_1 \cdot p_3 \cdot q_5 \cdot p_4 \cdot q_3 \cdot q_4$

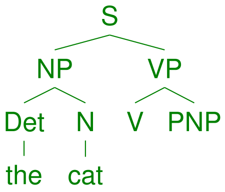
M. Rajman & J.-C. Chappelier

Notations (1/2)

For a context-free grammar $\mathcal{G}$, let:
$\mathcal{L}(\mathcal{G})$ the language recognized by $\mathcal{G}$
$\mathcal{R}(\mathcal{G})$ the set of rules of $\mathcal{G}$
$\mathcal{A}(\mathcal{G})$ the set of **partial** trees of $\mathcal{G}$
$\mathcal{T}(\mathcal{G})$ the set of complete trees of $\mathcal{G}$ (with root S, top-level symbol) $(\mathcal{T}(\mathcal{G}) \subset \mathcal{A}(\mathcal{G}))$

For a tree T of $\mathcal{A}(\mathcal{G})$:
$F(T)$ the left-ordered sequence of its leaves, and
$\mathrm{lmnt}(T)$ the left-most non-terminal leaf of T.
If T does not have any non-terminal leaf, $\mathrm{lmnt}(T) = \varepsilon$.

Example:

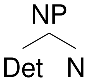


$F(T) = \{$ the, cat, V, PNP $\}$
and $\mathrm{lmnt}(T) = \mathrm{V}$

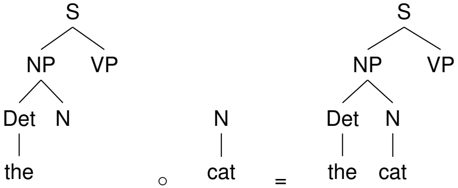
Notations (2/2)

Furthermore, the same notation R will be used for both the rule and the corresponding elementary tree:

$$\text{NP} \rightarrow \text{Det N}$$



The symbol $\circ$ denotes the internal composition rule on $\mathcal{A}(\mathcal{G})$ that returns the tree resulting from the substitution of the left-most non-terminal leaf of the left tree by the right tree when it is possible, and ε if not.



For a rule R of $\mathcal{R}(\mathcal{G})$, left(R) denotes the left-hand side of R

SCFG

Disambiguation: Let $\mathcal{G}$ be a Stochastic CFG and $W = w_1^n$ a sentence with several parse trees $T_1, ..., T_k$ according to $\mathcal{G}$. The goal is to choose among the T_is.

In a standard approach, such a choice is made on semantic/pragmatic criteria.

In the probabilistic approach, the choice is made according to the probabilities of the T_i trees. In other terms, we are looking for:

$$T = \underset{T_i \supset W}{\operatorname{argmax}} P(T_i | W)$$

But $P(T_i|W) = \frac{P(T_i, W)}{P(W)} = \frac{P(T_i)}{P(W)}$ since T_i precisely is a tree that analyses W

We are therefore looking for $T = \underset{T_i \supset W}{\operatorname{argmax}} P(T_i)$

Note: "$T_i \supset W$" means "$T \in \mathcal{T}(\mathcal{G}) : F(T) = W$"

M. Rajman & J.-C. Chappelier

SCFG: formalization

T_i is interpreted as the result of a given (unknown) stochastic process ξ

- ☞ because of the one-to-one mapping that exists in CFG between trees and derivations (sequences of rules), ξ is supposed to be a stochastic process on **rules**, i.e a random sequence in $\mathcal{R}(\mathcal{G})$

☞ We will therefore characterize $P(T)$ using $P(\xi = R_0, ..., R_n)$

$$P(\xi = R_0, ..., R_n) = P(R_0) \cdot \prod_{i=1}^{n} P(R_i | R_0, ..., R_{i-1})$$

Definition of the generating stochastic process

To fully define ξ, we need the definition of $P(R_0)$ and $P(R_i | R_0, ..., R_{i-1})$:

- R_0 is the *constant* "random" variable S (null-depth tree with root S, the start-symbol)
  Therefore $P(R_0 = \mathrm{S}) = 1$
- $P(R_i | R_0, ..., R_{i-1})$ is null if $\mathrm{left}(R_i) \neq \mathrm{lmnt}(R_0 \circ ... \circ R_{i-1})$

☞ What value for the probabilities that are not null?

M. Rajman & J.-C. Chappelier

Value for $P(R_i|R_0,\ldots,R_{i-1})$

As up to now, this probability is conditioned by $\text{left}(R_i) = \text{lmnt}(R_0 \circ \ldots \circ R_{i-1})$
If we make the assumption that it is conditioned **ONLY** by this, then

$$P(R_i|R_0,\ldots,R_{i-1}) = P(R_i|\text{lmnt}(R_0 \circ \ldots \circ R_{i-1})) = P(R_i|\text{left}(R_i))$$

which therefore only depends on R_i and will be denoted by $p(R_i)$. It is called the "*stochastic coefficient*" of the rule R_i
☞ $p(R_i)$ is a **parameter** of the processus ξ and, by construction, we have:

$$\forall R \in \mathcal{R}(\mathcal{G}) \quad \sum_{R' \in \mathcal{R}(\mathcal{G}):\text{left}(R')=\text{left}(R)} p(R') = 1$$

Notice that limiting $P(R_i|R_0 \ldots R_{i-1})$ to the conditioning by $P(R_i|\text{lmnt}(R_0 \circ \ldots \circ R_{i-1}))$ only is a **strongly restrictive hypothesis** on the process.

Probability of a tree? (1/2)

Finaly, the probability of a (valid) sequence of rules is:

$$P(R_0, ..., R_n) = \prod_{i=1}^{n} p(R_i)$$

Each T in $\mathcal{T}(\mathcal{G})$ corresponds to a unique (valid) sequence of rules, therefore

$$P(T) = P(R_1, ..., R_k) = \prod_{i=1}^{k} p(R_i)$$

In short: For SCFGs, the probability of a tree is the product of the stochastic coefficient associated to its rules

M. Rajman & J.-C. Chappelier

Probability of a tree? (2/2)

BUT... is it really a probabilty on $\mathcal{T}(\mathcal{G})$?...

What is $\displaystyle\sum_{T \in \mathcal{T}(\mathcal{G})} P(T)$?

- It converges (increasing and upper-bounded by 1)
- towards a limit lower or equal to 1
- But that can be < 1

Example: S $\rightarrow$ S S (p) S $\rightarrow$ a (1-p)

☞ $\displaystyle\sum_{T \in \mathcal{T}(\mathcal{G})} P(T) = \min\left(1, \frac{1-p}{p}\right)$

Therefore the correct probabilization is: $\displaystyle\widehat{P}(T) = \frac{P(T)}{\sum_{T \in \mathcal{T}(\mathcal{G})} P(T)}$

In the case where the grammar is **consistent** (i.e. $\sum P(T) = 1$), or in the case where only the maximum probability is considered, the two approches are equivalent.
The only problematic case here is when one deals simultaneously with several not consistent grammars.

Probability of a sentence $P(W)$

The probability of a sentence is defined by:

$$P(W) = \sum_{T_i \supset W} \widehat{P}(T)$$

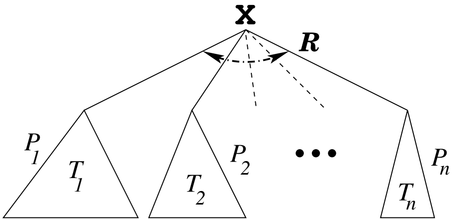
Notice that $P(T, W) = \widehat{P}(T) \cdot \delta(W = F(T))$ (Kronecker notation), which justifies the formulas used at the beginning of the lecture.

SCFG: implementation (1/2)

It is possible to compute $\operatorname{argmax} P(T_i)$ and/or $P(W) = \sum P(T_i)$ during the bottom-up phase of the CYK analysis, using dynamic programming.

For a given element in a cell, a value v_i representing the maximum (or the sum) of the probabilities of its interpretations is stored.

Notice: if T is



then $$\begin{aligned} P(T) &= \prod p(R_i) \\ &= p(R) \cdot P_1 \cdots P_n \end{aligned}$$

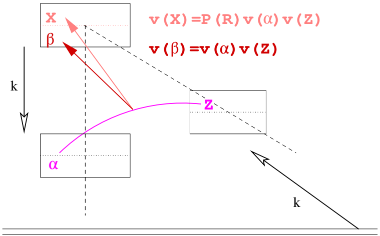
M. Rajman & J.-C. Chappelier

SCFG: implementation (2/2)

When a new interpretation of element A (be it a non-terminal X or an item $[\beta \bullet \cdots]$) is built by the composition of elements B and C), the value v_A is updated according to:

(when computing the max) $v_A = \max(v_A, v_B\, v_C\, \rho_A)$

(or, when computing the sum) $v_A = v_A + v_B\, v_C\, \rho_A$

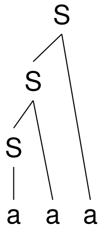


with $\rho_A = 1$ if element A is an item $[\beta \bullet \ldots]$
and $\rho_A = p(R)$ if element A is a non-terminal X, obtained by applying rule R

The initial value for the v_As is 0

SCFG: implementation example

$$
\begin{array}{lll}
S \rightarrow S\ S & (0.1) \\
S \rightarrow a\ S & (0.2) \\
S \rightarrow S\ a & (0.3) \\
S \rightarrow a & (0.4)
\end{array}
$$



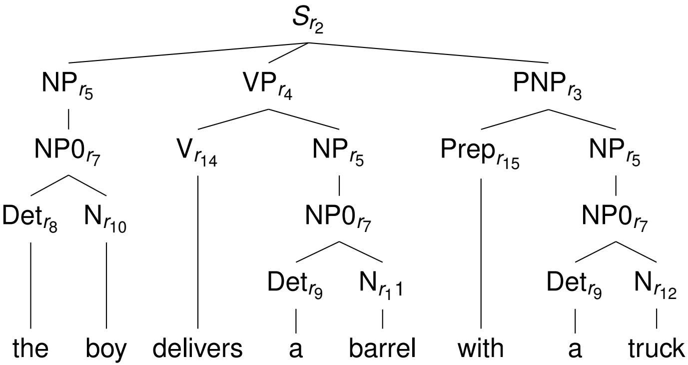
S $(0.3 \times 0.3 \times 0.4)$		
S (0.3×0.4)	S (0.3×0.4)	
S (0.4)	S (0.4)	S (0.4)
a	a	a

M. Rajman & J.-C. Chappelier
EPFL

Grammar extraction from a treebank (1/3)

Consider a treebank made of the two following parse trees:

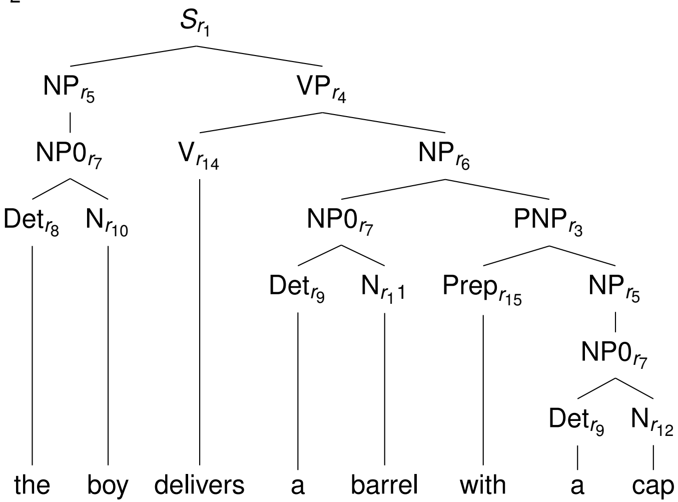
T_1:



M. Rajman & J.-C. Chappelier
EPFL

Grammar extraction from a treebank (2/3)

T_2:



M. Rajman & J.-C. Chappelier
EPFL

Grammar extraction from a treebank (3/3)

From the trees present in the corpus, we can extract the context-free grammar G, made of the following 15 rules:

rule	p_i
r_1: S -> NP VP	p_1
r_2: S -> NP NP PNP	p_2
r_3: PNP -> Prep NP	p_3
r_4: VP -> V NP	p_4
r_5: NP -> NP0	p_5
r_6: NP -> NP0 PNP	p_6
r_7: NP0 -> Det N	p_7

rule	p_i
r_8: Det -> the	p_8
r_9: Det -> a	p_9
r_{10}: N -> boy	p_{10}
r_{11}: N -> barrel	p_{11}
r_{12}: N -> truck	p_{12}
r_{13}: N -> cap	p_{13}
r_{14}: V -> delivers	p_{14}
r_{15}: Prep -> with	p_{15}

where the p_i denote the probabilities associated with each of the rules

☞ How can we estimate them?

M. Rajman & J.-C. Chappelier

Estimating the probabilities

supervised learning: When a tree-bank (annotated corpus) is available, stochastic coefficients are estimated by the relative frequencies (e.g. maximum likelihood estimation:

$$p(R) = \frac{\text{nb. occurrences of } R}{R' \text{ such that left}(R') = \text{left}(R)}$$

or with some smoothing (prefered))

unsupervised learning: When only text is available (**and** also a grammar) : EM estimation of the coefficients : inside-outside algorithm

- iterative algorithm
- converges towards a local minimum
- highly sensitive to initial values

hybrid approaches: using a (small) tree-bank and a (large) corpus of text

M. Rajman & J.-C. Chappelier
EPFL

Estimating the probabilities: example

In our case (supervised learning), with MLE, we get:

rule	p_i
r_1: `S -> NP VP`	1/2
r_2: `S -> NP NP PNP`	1/2
r_3: `PNP -> Prep NP`	1
r_4: `VP -> V NP`	1
r_5: `NP -> NP0`	5/6
r_6: `NP -> NP0 PNP`	1/6
r_7: `NP0 -> Det N`	1

rule	p_i
r_8: `Det -> the`	1/3
r_9: `Det -> a`	2/3
r_{10}: `N -> boy`	1/3
r_{11}: `N -> barrel`	1/3
r_{12}: `N -> truck`	1/6
r_{13}: `N -> cap`	1/6
r_{14}: `V -> delivers`	1
r_{15}: `Prep -> with`	1

M. Rajman & J.-C. Chappelier
EPFL

Keypoints

- Probabilities in syntax are a numerical representation of implicit linguistic constraints used to measure the adequation between the sentence and the model
- The role of probabilities is to identify the correctness of the sentence and eventually to choose one interpretation among several
- SCFG fundamentals:
  - $\sum_{R':\mathrm{left}(R')=\mathrm{left}(R)} p(R') = 1$
  - $P(T) = \prod_{i=1}^{n} p(R_i)$
- SCFG limitation: $P(R_i|R_0,...,R_{i-1}) = P(R_i|\mathrm{left}(R_i))$
- SCFG may be inconsistent
- Calculation of probabilities of syntactic interpretations of sentences (in case of SCFGs)
- Estimation of probabilities of SCFGs from training corpora

M. Rajman & J.-C. Chappelier

References

[1] C. D. Manning, H. Schütze, *Foundations of Statistical Natural Language Processing*, ch. 11, 12, MIT, 1999.

[3] D. Jurafsky & J. H. Martin, *Speech and Language Processing*, ch. 12, Prentice Hall, 2000.

[4] R. Dale, H. Moisl & H. Sommers, *Handbook of Natural Language Processing*, ch. 22, Dekker, 2000.